

João Antonio Campos Panceri

Reconhecimento de Expressões Faciais Baseado em Active Appearance Model

Brasil

2017

João Antonio Campos Panceri

Reconhecimento de Expressões Faciais Baseado em Active Appearance Model

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Universidade Federal do Espírito Santo – UFES

Departamento de Engenharia Elétrica

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Dr. Evandro Ottoni Teatini Salles

Brasil

2017

João Antonio Campos Panceri
Reconhecimento de Expressões Faciais Baseado em Active Appearance Model/
João Antonio Campos Panceri. – Brasil, 2017-
76 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Evandro Ottoni Teatini Salles

Dissertação (Mestrado) – Universidade Federal do Espírito Santo – UFES
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica, 2017.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III.
Faculdade de xxx. IV. Título

CDU 02:141:005.7

*Este trabalho é dedicado a minha família em especial a minha futura esposa Raquel,
meus queridos pais Jonas e Joana e aos meus irmãos Rafael e Renata.*

Agradecimentos

Os agradecimentos são direcionados ao meus estimados mestres que me ajudaram a chegar até aqui, em especial Dr. Marco Antônio Quadros Leite de Souza e Dr. Daniel Cavalieri professores da minha graduação e ao meu orientador de mestrado Dr. Evandro Ottoni Teatini Salles. A esses meu eterno respeito e admiração. Almejo um dia atingir o nível de conhecimento e profissionalismo desses notáveis professores. Não poderia deixar de prestar agradecimentos ao meu admirado amigo Gênesis Lopes por seu apoio e colaboração.

Pelo incentivo, suporte e financiamento à pesquisa, agradeço ao CNPq e ao PPGEE-UFES.

*"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito."
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

Este trabalho propõe uma solução de um sistema capaz de classificar, segundo os critérios estabelecidos por Paul Ekman, expressões básicas de comportamento na face humana em imagens digitais. A primeira etapa é referente a delimitação da expressão facial na imagem, para isto será utilizado o método baseado no algoritmo de modelagem AAM (Active Appearance Model) que possui uma exímia capacidade de ajustar um conjunto de pontos cuja a uma forma se adeque a uma face. Com intuito de realizar o processo de classificação, são extraídas as características tanto da imagem com expressão definida quanto da face com expressão neutra, então a diferença entre as características das duas imagens são aplicadas a um classificador de padrões como *Support Vector Machine* (SVM) e Rede Neural com o objetivo de classificar a expressão em questão. Por fim, para a validação do algoritmo e testes comparativos serão realizados simulações utilizando o banco de dados de expressões faciais amplamente conhecido na literatura Cohn-Kanade (CK+). Os resultados sugerem que é possível associar as deformações musculares, causadas pelas expressões faciais, com a triangulação de Delaunay obtida a partir das marcações ajustadas pela técnica AAM.

Palavras-chave: Expressões Faciais, Active Appearance Model, Support Vector Machine, Rede Neural, Reconhecimento de Padrões.

Abstract

This work proposes the solution of a system capable of classifying, according to Paul Ekman's principles, basic behavior expressions in a human face on a digital image. The first step refers to the outlining of the expression on the image. For this, a method based on the modeling algorithm AAM (Active Appearance Model) is applied. This method has an exceptional ability of adjusting a group of points whose shapes can be proper for a face. In order to initiate the classification process, the characteristics of the image with the defined expression, as well as the neutral expression one, are extracted, and the differences between them a classifier as Support Vector Machine and Neural Network performs the process of recognizing the expression. Finally, to authenticate the algorithm and the comparing tests, simulations with the facial expressions data bank, widely known as Cohn-Kanade (CK+), will be used. The results suggest that it is possible to associate muscular deformations, caused by facial expressions, with the Delaunay triangulation of landmarks reached by fitting AAM technique.

Keywords: Facial Expressions, Active Appearance Model, Support Vector Machine, Neural Network, Pattern Recognition.

Lista de ilustrações

Figura 1 – Sistema FACS - Unidades de ação (UAs). Fonte: https://www.eilert-akademie.de/	17
Figura 2 – Sistema FACS - Exemplo de classificação pelo método FACS. Fonte: Modificada a partir de https://www.eilert-akademie.de/	18
Figura 3 – Ajuste AAM. Fonte: Modificada a partir do banco de dados CK+.	19
Figura 4 – Esquema simplificado do processo de identificação de expressões faciais. Fonte: Própria.	19
Figura 5 – Diagrama de blocos da arquitetura proposta. Fonte: Própria.	20
Figura 6 – Marcações (<i>Landmarks</i> .) Fonte: Modificada a partir do banco de dados CK+.	24
Figura 7 – <i>Procrustes Analysis</i> - Dados de entrada. Fonte: Própria.	25
Figura 8 – <i>Procrustes Analysis</i> - Dados tratados. Fonte: Própria.	27
Figura 9 – <i>Procrustes Analysis</i> - Forma média. Fonte: Própria.	27
Figura 10 – Energia dos autovetores. Fonte: Própria.	28
Figura 11 – Etapas para a construção do modelo de forma. Fonte: Própria.	28
Figura 12 – Triangulação de Delaunay. Fonte: Própria.	29
Figura 13 – Triangulação de Delaunay - <i>Flip</i> . Fonte: Própria.	30
Figura 14 – Triangulação de Delaunay. Fonte: Modificada a partir do banco de dados CK+.	30
Figura 15 – Função de Mapeamento - <i>Piecewise Affine Warp</i> . Fonte: (MATTHEWS; BAKER, 2004).	31
Figura 16 – Textura média. Fonte: Própria.	33
Figura 17 – Energia dos autovetores - Conjunto de Textura. Fonte: Própria.	34
Figura 18 – Etapas para a construção do modelo de textura. Fonte: Própria.	34
Figura 19 – Algoritmo AAM. Fonte: (MATTHEWS; BAKER, 2004).	35
Figura 20 – Diagrama de blocos descrevendo a metodologia de criação do modelo AAM. Fonte: Própria.	35
Figura 21 – Representação gráfica simplificada do algoritmo k-nn. Fonte: Modificada a partir do banco de dados CK+.	40
Figura 22 – Modelo de um neurônio. Fonte: (HAYKIN, 2009)	41
Figura 23 – Função sigmóide com parâmetro de inclinação a variável. Fonte: Própria.	42
Figura 24 – Hiperplanos separando duas classes a partir de margens iguais. Fonte: (THEODORIDIS; KOUTROUMBAS, 2009).	43
Figura 25 – Os candidatos a hiperplano canônico. Fonte: (THEODORIDIS; KOUTROUMBAS, 2009).	45

Figura 26 – Parâmetro C de tolerância a erros . Fonte: (THEODORIDIS; KOU-TROUMBAS, 2009).	45
Figura 27 – Classificação SVM multiclasse. Fonte: (PEDROSO; SALLES, 2012). .	46
Figura 28 – Sequencia de imagens da base CK+ contendo a evolução da expressão facial surpresa. Fonte: Modificada a partir do banco de dados CK+. . .	48
Figura 29 – Banco de dados utilizado: JAFFE - Japanese Female Facial Expression. Fonte: Modificada a partir do banco de dados JAFFE.	48
Figura 30 – Evolução da expressão facial. Fonte: Modificada a partir do banco de dados CK+.	50
Figura 31 – Gráfico da diferença entre as áreas de uma expressão de surpresa e neutra - Indivíduo 1. Fonte: Modificada a partir do banco de dados CK+. .	51
Figura 32 – Gráfico da diferença entre as áreas de uma expressão de surpresa e neutra - Indivíduo 2. Fonte: Modificada a partir do banco de dados CK+. .	52
Figura 33 – Gráfico da diferença entre as áreas de uma expressão de feliz e neutra - Indivíduo 1. Fonte: Modificada a partir do banco de dados CK+. . . .	53
Figura 34 – Representação do método de validação <i>Cross-Validation</i> com 10 folds. Fonte: Própria.	54
Figura 35 – Representação de uma rede neural com múltiplas camadas. Fonte: Própria. .	54
Figura 36 – Melhor resultado AAM - Erro mínimo: 1,5 pixels. Fonte: Própria. . . .	55
Figura 37 – Pior resultado AAM - Erro máximo: 15,14 pixels. Fonte: Própria. . . .	56
Figura 38 – Erro médio entre as <i>landmarks</i> . Fonte: Própria.	56
Figura 39 – Distribuição dos dados em 3 dimensões. Fonte: Própria.	57
Figura 40 – Projeção dos dados em 3 dimensões - JAFFE <i>Database</i> . Fonte: Própria. .	60
Figura 41 – Projeção dos dados realizada por Pedroso - JAFFE <i>Database</i> - Fonte: (PEDROSO; SALLES, 2012).	61
Figura 42 – Projeção dos dados a partir do conceito de descritor de (PEDROSO; SALLES, 2012) para a base CK+. Fonte: Própria.	62
Figura 43 – Demonstração gráfica da redução de dimensionalidade através da PCA	72
Figura 44 – Esquema das etapas do PCA	72

Lista de tabelas

Tabela 1 – Configuração dos classificadores.	54
Tabela 2 – Acurácia por classificador com validação 10-fold - Marcações obtidas a partir do ajuste AAM na base CK+.. . . .	58
Tabela 3 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações obtidas a partir do ajuste AAM na base CK+.	58
Tabela 4 – Acurácia por classificador com validação 10-fold - Marcações <i>Ground truth</i> da base CK+.. . . .	59
Tabela 5 – Acurácia em percentagem (%) a partir da perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações <i>Ground Truth</i> da base CK+ com classificador SVM 10-fold.	59
Tabela 6 – Acurácia por classificador com validação 10-fold - Marcações obtidas a partir do ajuste AAM na base JAFFE.	61
Tabela 7 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações obtidas a partir do ajuste AAM à base JAFFE.	62
Tabela 8 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor de (PEDROSO; SALLES, 2012) aplicado as marcações obtidas a partir do ajuste AAM à base CK+.. . . .	63
Tabela 9 – Resultado de trabalho correlatos.	63

Lista de abreviaturas e siglas

AAM	<i>Active Appearance Model</i>
FACS	<i>Facial Action Coding System</i>
UA	Unidades de Ação
PCA	Análise de Componentes Principais
k-nn	<i>k-Nearest Neighbour</i>
MPL	<i>Multi-Layer Perceptrons</i>
CK+	Cohn-Kanade <i>Database</i>
JAFFE	<i>The Japanese Female Facial Expression Database</i>
CNN	<i>Convolution Neural Network</i>
SVM	<i>Support Vector Machine</i>

Sumário

1	INTRODUÇÃO	14
1.1	<i>Facial Action Coding System (FACS)</i>	16
1.1.1	Caracterização do Problema e Hipótese	17
1.2	Objetivo	19
1.2.1	Arquitetura	19
1.3	Metodologia de Avaliação dos Resultados	20
1.4	Estado da Arte	21
1.5	Organização do Texto	23
2	<i>ACTIVE APPEARANCE MODEL - AAM</i>	24
2.1	Definição	24
2.2	Construção do Modelo	24
2.2.1	Construção do Modelo de Forma	25
2.2.1.1	<i>Procrustes Analysis</i>	25
2.2.1.2	Modelo de Forma - (<i>Shape Model</i>)	26
2.2.2	Construção do Modelo de Textura	29
2.2.2.1	Triangulação de Delaunay	29
2.2.2.2	<i>Piecewise Affine Warp</i>	30
2.2.2.3	Modelo de Textura - (<i>Appearance Model</i>)	32
2.2.3	Modelo Completo e Ajuste AAM	33
2.3	Solução AAM	35
2.3.1	Fast-SIC	36
3	CLASSIFICADORES	39
3.1	<i>k-Nearest Neighbour - (k-nn)</i>	39
3.2	Rede Neural Multicamada	40
3.3	SVM - Support Vector Machine	42
3.3.1	Classes Linearmente Separáveis	42
3.3.2	O parâmetro de penalidade C	44
3.3.3	SVM para Multiclasses	45
4	METODOLOGIA	47
4.1	Bancos de Dados	47
4.1.1	Cohn-Kanade (CK+) <i>Database</i>	47
4.1.2	<i>The Japanese Female Facial Expression (JAFPE) Database</i>	47
4.2	Sistema Proposto	49

4.2.1	Construção do Modelo AAM	49
4.2.2	Extrator de Características	50
4.3	Ambiente de Desenvolvimento	52
4.3.1	Ajuste dos Classificadores	52
5	RESULTADOS	55
5.1	Ajuste AAM	55
5.2	Extração de Características	55
5.3	Classificação dos Dados	57
5.4	Comparação com outros trabalhos	59
6	CONCLUSÕES	64
6.1	Trabalhos Futuros	65
	Referências	67

APÊNDICES 70

APÊNDICE A – PCA - ANÁLISE DE COMPONENTES PRINCIPAIS

A.1	Matriz de Covariância	73
A.2	Autovalores e Autovetores	75
A.3	Autovalores e Autovetores	75
A.4	Matriz de Transformação de Hotelling	75

1 Introdução

Nas relações humanas, as emoções possuem um papel preponderante durante todas as etapas do ciclo vital. Seja nos relacionamentos íntimos, nas interações familiares, no trabalho ou na escola, elas auxiliam a adaptação às diversas situações da vida (REEVES, 2006).

A tarefa de conhecer as emoções é uma habilidade complexa, sendo o reconhecimento das emoções, e em específico das expressões emocionais em faces humanas, um dos componentes centrais para utilização saudável das mesmas (IZARD, 2001).

A comunicação humana consiste na materialização do pensamento e/ou sentimento em signos conhecidos pelas pessoas envolvidas. Estes símbolos são transmitidos pelo(s) emissor(es) e posteriormente reinterpretadas pelo(s) receptor(es), e tem por finalidade expressar pensamentos, ideias e sentimentos que possam ser compreendidos. A comunicação pode ocorrer de duas maneiras distintas:

1. Verbal: Abrange tanto a comunicabilidade oral quanto a escrita, neste tipo de comunicação a clareza das mensagens passadas está relacionada à compatibilidade do acervo linguístico e intelectual dos envolvidos na troca de informações. É portanto, a forma de comunicação mais adequada para transmitir ideias de maior complexidade. A comunicação verbal oral tem como característica precípua o fato das mensagens compartilhadas entre os interlocutores poderem ser transmitidas através do vocábulo em concomitância à linguagem não-verbal. A comunicação verbal escrita, por sua vez, foi um grande avanço, já que nas sociedades ágrafas o conhecimento era passado de forma oral, e isto era um empecilho na transmissão do conhecimento por causa da limitação natural da memória humana.

2. Não-verbal: Na linguagem não verbal a comunicação geralmente é efetuada através de sinais visuais como expressão facial, dança, pintura, gestos, postura, mímica, bandeira, sinais de trânsito, entre outros; em outras palavras a troca de informações à qual não se utiliza de um vocábulo para se comunicar. De acordo com Rogers e Steinfatt (1999) a comunicação não-verbal está presente a todo o momento nas relações pessoais, já que a mesma pode ser inconsciente e não intencional. Para que o processo de comunicação não-verbal tenha sentido e possa ser interpretado, é necessário que o destinatário da mensagem reconheça os sinais (gestos, acenos, expressões faciais, entre outros) emitidos pela linguagem corporal durante o processo de comunicação, sendo o emissor da mensagem consciente ou não dos sinais que emite (FURNHAM, 2001).

Vale a pena ressaltar o fato de ao longo da história a humanidade ter desenvolvido mecanismos cada vez mais sofisticados para auxiliar e potencializar o envio e recepção das mensagens verbais e não-verbais. O avanço dos meios de comunicação se deu forma ainda

mais relevante nos últimos séculos devido ao desenvolvimento de novas tecnologias na área de telecomunicações.

O cérebro humano é formado por dois hemisférios, o lado esquerdo é encarregado de controlar a linguagem verbal e as capacidades lógico-analíticas, enquanto o lado direito é responsável por mobilizar uma resposta não-verbal, destacando-se em tarefas visuais e motoras. As manifestações da linguagem verbal oral e da linguagem não-verbal são complementares em seu processo global, tornando a comunicação humana mais rica, compreensível e acessível (BIRCK, 2008).

O corpo fala, se manifesta, se expressa, transmite mensagens, não apenas com palavras, mas através de expressões faciais (olhos, lábios) gestos com as mãos, postura física, ritmo do corpo (caminhar, correr), perpassando informações muitas vezes importantes sobre si, sem utilizar-se das palavras (linguagem verbal). Os gestos fazem parte dos recursos de comunicação, como uma palavra ou uma frase, pois expressa suas emoções e, também algumas vezes, pode contradizer as palavras (BIRCK, 2008).

Segue os tipos de comunicações definidas por Birck :

- paralinguagem: modificação das características sonoras da voz;
- proxêmica: uso e organização do espaço físico;
- tacêsica: linguagem do toque;
- cinésica: linguagem do corpo;
- características físicas: forma e aparência do corpo.

Dentre as cinco apresentadas, a cinésica assume um papel imprescindível na decodificação das mensagens recebidas durante as interações interpessoais. A linguagem corporal é uma forma complexa de interação, da qual temos pouca consciência, ocorrendo corriqueiramente fora do nosso controle.

As expressões faciais é o meio pelo qual o ser humano possui, através da linguagem corporal, a maior comunicabilidade das emoções. Ela é o resultado de determinadas configurações ou contrações dos músculos faciais que provocam modificações e deformações na face. Estas alterações possuem intrinsecamente padrões universais, por isso a identificação de emoções pela face pode ser uma ferramenta útil para proporcionar uma melhor interação entre humanos e sistemas inteligentes. A classificação automática expressões faciais pode ser utilizada em diversas áreas e aplicações:

- Ambiente de trabalho: monitoramento/análise do estado emocional dos funcionários ao decorrer do expediente. Nesse caso, a expressão facial pode se tornar um indicativo

institucional em qual área há necessidade da implantação melhorias para saúde emocional dos empregados;

- Portadores de necessidades especiais: detectar automaticamente comandos de modo a auxiliar pessoas com dificuldades de locomoção motora severa (FARIA RODRIGO A. M. BRAGA, 2007);
- Ensino a distância: o mapeando das emoções do aluno selecionando o melhor método de ensino e material didático aplicado;
- Saúde: monitoramento de pacientes e auxílio em diagnósticos médicos (RUSSELL; FERNÁNDEZ-DOLS, 1997);
- Computação Gráfica: criação e manipulação de avatares digitais (DAVIES; DALTON; CAMPBELL, 2011);
- Análise da resposta de consumidores a um terminado conteúdo. (MOON; SHARMA; JUNG, 2012).

1.1 *Facial Action Coding System (FACS)*

A primeira publicação de grande relevância sobre a expressão facial das emoções foi escrita por Charles Darwin em sua obra "The Expression of the Emotions in Man and Animals" (DARWIN, 1872). Darwin propôs pautado por justificativas evolucionistas e heranças genéticas que as expressões das emoções são universais, isto é, iguais para a espécie, independentemente de sua cultura particular em questão. Em contraposição a Darwin, a antropóloga Margaret Mead (1967) afirmou que a expressão facial das emoções teria modificações de acordo ao ambiente cultural no qual o indivíduo estivesse inserido.

Paul Ekman, com intuito de aferir quais das teorias estava correta, baseou-se nos estudos das teorias emoções de seu orientador e parceiro de pesquisa S. S. Tomkins. Utilizou como estudo de caso uma tribo da Nova Guiné, que nunca havia tido contato com outras civilizações e tecnologias modernas. Além disso, tinham praticamente desconhecimento do próprio rosto, tendo em vista que não possuíam espelhos ou tinham rios que os permitissem ver o próprio reflexo na água. Foi neste cenário que possibilitou a Ekman encontrar evidências necessárias para constatar que as emoções são expressas na face de forma universal (EKMAN; FRIESEN, 1971).

Em 1971 Paul Ekman, W. Friesen e S. S. Tomkins, elaboraram um método de medição das ações faciais *Facial Affect Scoring Technique* (FAST) (EKMAN; FRIESEN, 1971). Em (EKMAN; FRIESEN, 1978) o método passou por modificações e foi renomeado como *Facial Action Coding System* (FACS).

A teoria *Facial Action Coding System* (FACS) desenvolvida em (EKMAN; FRIESEN, 1978) é um sistema abrangente, com base anatômica. Nela, busca-se descrever todos os possíveis movimentos faciais em humanos. A teoria FACS discrimina quarenta e quatro Unidades de Ação (UA's), exemplificadas na Figura 2. Cada UA tem seu código numérico, enumeradas de 1 a 44, e seus respectivos músculos que quando ativados produzem um dado movimento da face. Como por exemplo: UA1 significa o levantamento da sobrancelha interna, UA2 o levantamento da sobrancelha externa). A Figura 3 demonstra um exemplo de classificação de expressões faciais através do método FACS. A tabela completa com as associações das UA's sistematizadas para as expressões básicas definidas por Ekman (surpresa, medo, tristeza, felicidade, nojo e nervoso) pode ser vista em (EKMAN; FRIESEN, 1978).

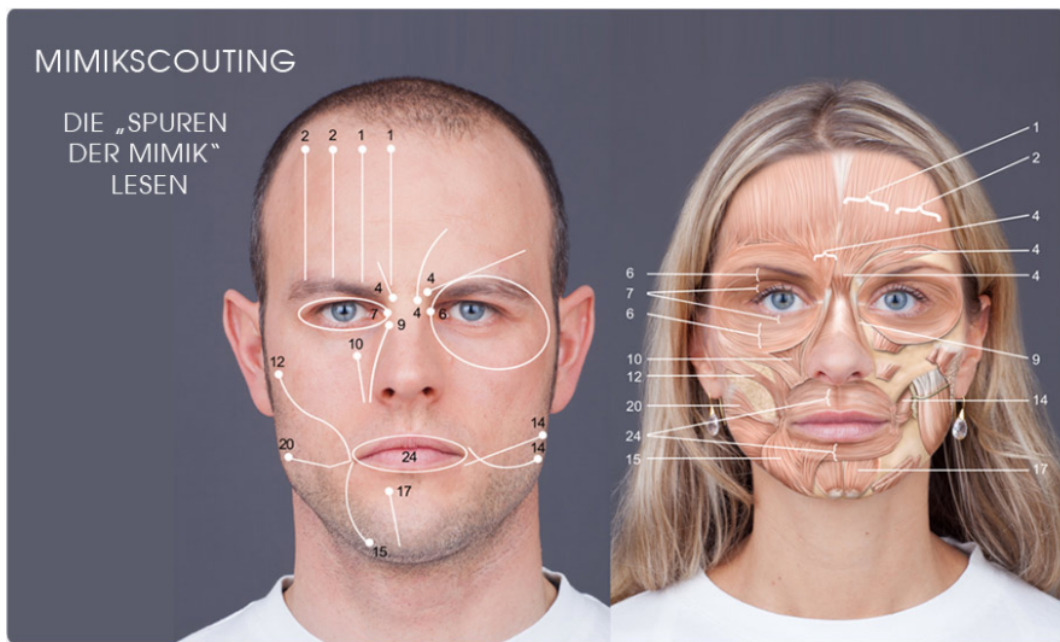


Figura 1 – Sistema FACS - Unidades de ação (UAs). Fonte: <https://www.eilert-akademie.de/>.

1.1.1 Caracterização do Problema e Hipótese

A aplicação do método FACS é eficaz como indicador de expressão facial. Contudo encontrar computacionalmente as AU's é uma tarefa complexa, visto que é muito difícil detectar a musculatura da face humana através de uma imagem digital. Esse problema ocorre, basicamente, por duas razões, a primeira se deve a complexidade da identificação automática da posição dos músculos, e a segunda dificuldade está em referenciar a posição estimada do músculo em relação ao seu estado anterior.

Por este motivo um método computacional de identificação de pontos específicos em uma face possibilita uma ferramenta de estimação de posição. Entretanto tal propósito



Figura 2 – Sistema FACS - Exemplo de classificação pelo método FACS. Fonte: Modificada a partir de <https://www.eilert-akademie.de/>.

não se trata de uma tarefa trivial, sobretudo quando é necessário um número elevado de pontos.

Outra dificuldade encontrada é que mesmo possuindo a posição dos pontos chave em uma face, ainda é difícil avaliar que os mesmos indicam ativação de alguma AU. A título de exemplo, a distância entre os pontos das extremidades horizontais da boca é insuficiente para determinar com precisão um sorriso, posto que para cada indivíduo poderá haver limiar, ou seja, uma distância específica que depende das medidas da boca para indicar um sorriso.

Uma ferramenta que seja capaz de mapear pontos específicos em uma imagem de uma face seria um instrumento fundamental para o propósito de identificar expressões faciais, uma vez que a posição das AU's poderão ser estimadas. Nesse aspecto, a ferramenta *Active Appearance Model* AAM apresenta uma potencial capacidade de modelar objetos através de um ajuste fino de pontos. Em outras palavras, mesmo apresentando variações em sua forma e pose, o método AAM é capaz de encontrar um conjunto de pontos, que podem compor triângulos cuja a forma se ajusta ao objeto em questão, conforme a Figura 4 demonstra.

Assim, tendo em vista a dificuldade já mencionada a respeito da aplicação do FACS em imagens computacionais, nesse trabalho propõe-se a hipótese que os triângulos obtidos a partir dos pontos citados acima é uma estratégia de aproximação das AU's estudadas por Paul Ekman.

Com o objetivo de solucionar a falta de referencial, pretende-se demonstrar que utilizando a face neutra como estado inicial é possível analisar alterações espaciais em uma face com unidades de ação (AUs) ativadas.

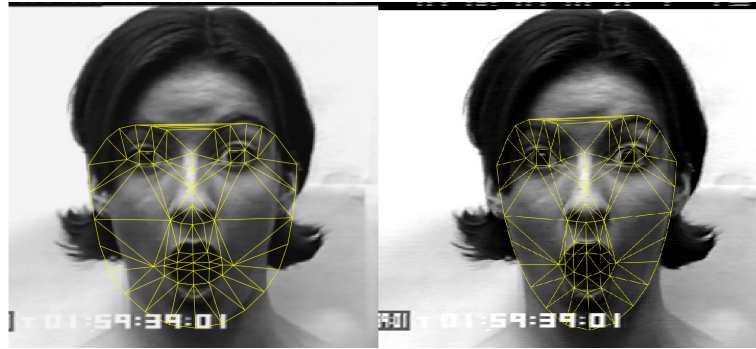


Figura 3 – Ajuste AAM. Fonte: Modificada a partir do banco de dados CK+.

1.2 Objetivo

O principal objetivo deste trabalho é propor uma solução de um sistema capaz de identificar expressões de comportamento não-verbal na face humana em imagens digitais usando-se o AAM. Utilizando uma ferramenta de modelagem de objetos pretende-se encontrar uma máscara de pontos que tome a forma de uma face ainda que a mesma tenha sido alterada por uma expressão facial. E que por intermédio do conjunto de pontos obtidos seja possível classificar a expressão em uma das emoções básicas.

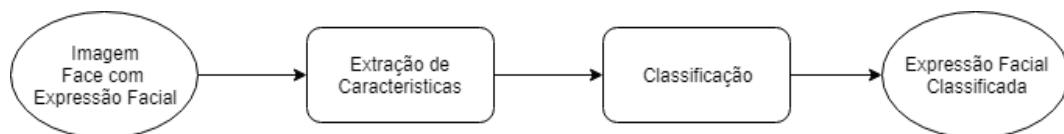


Figura 4 – Esquema simplificado do processo de identificação de expressões faciais. Fonte: Própria.

1.2.1 Arquitetura

A arquitetura proposta é construída da seguinte forma: as entradas do sistema consistem em duas imagens contendo respectivamente uma face neutra e uma face com uma expressão facial. Em cada uma é inicializado uma máscara genérica de pontos obtida através da modelagem AAM. A partir de um processo de minimização os parâmetros do modelo são encontrados de modo a ajustar a máscara a face. Com pontos ajustados adequadamente, uma triangulação ótima é realizada, e então calcula-se a área de cada triângulo. O descritor de características da expressão analisada é concebido baseado na diferença entre as áreas dos triângulos correspondentes da face neutra e da face com unidade de ação ativada. Por fim, a expressão é classificada em uma das classes de treinamento. A arquitetura geral da abordagem proposta pode ser resumida como na Figura 4.

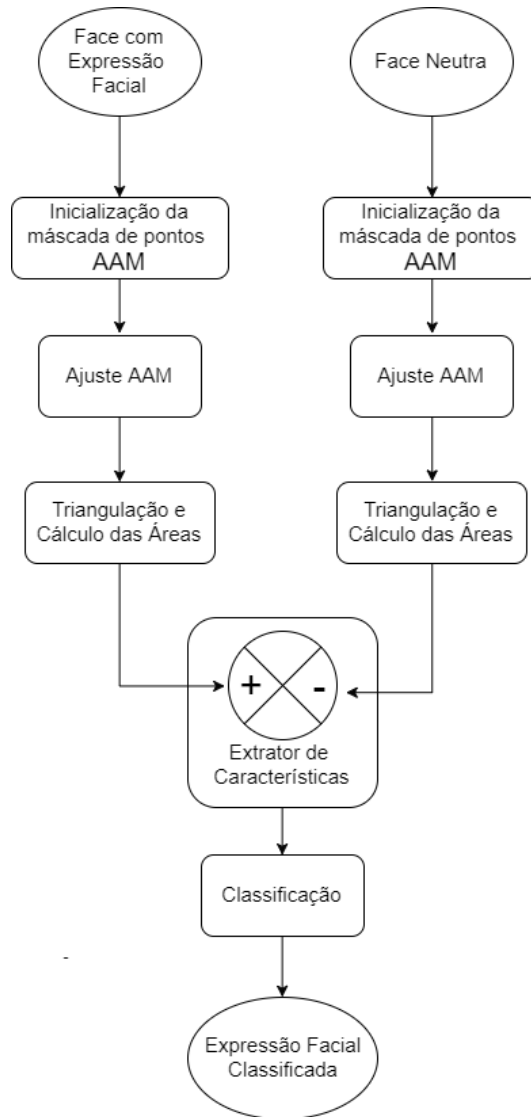


Figura 5 – Diagrama de blocos da arquitetura proposta. Fonte: Própria.

1.3 Metodologia de Avaliação dos Resultados

A primeira avaliação é analisar a capacidade de ajuste do modelo AAM em modelar faces. Tendo como base de comparação os pontos *Ground truth* fornecidos originalmente no banco de dados, pretende-se aplicar como metodologia de avaliação o cálculo do erro médio entre os pontos obtidos e pontos reais.

Outro ponto importante a se avaliar é a qualidade discriminativa do vetor descritor proposto, uma vez que a hipótese de similaridade entre ele e o método FACS precisa ser verificada. Para tal objetivo, os resultados obtidos na etapa de classificação e validação necessitam ser comparados com as informações corretas fornecidas pela base de dados.

1.4 Estado da Arte

Um dos primeiros estudos aplicados à análise computacional de expressões faciais foi desenvolvido por (FUJIMORA, 1978) tendo com base 20 pontos identificados manualmente em uma sequência de imagens. Posteriormente, juntamente com Paul Ekman (DONATO M. BARTLETT, 1999) aplicaram as técnicas *Local Feature Analysis* (LFA), *Independent Component Analysis* (ICA) para realizar computacionalmente a classificação de expressões faciais baseado na teoria FACS.

O filtro de Gabor também pode ser aplicado para extração de características de expressões faciais, como realizado por (DYER, 2005). Nesse caso, aplica-se a filtragem de Gabor por 3 escalas e 6 direções, produzindo-se o vetor descritor de textura. Com um classificador *Support Vector Machines* (SVM) o sistema apresentou uma acurácia de 63,3%. Em (OMLIN, 2006), utilizou-se características do tipo Haar e o classificador AdaBoost para a construção do descritor, demonstrando que as técnicas *Haar* e AdaBoost apresentam resultados equivalentes ao Gabor e SVM em relação ao desempenho alcançado. Entretanto, a combinação de *Haar* e AdaBoost demonstrou ser duas vezes mais rápida que o método proposto por (DYER, 2005).

Dentre algumas técnicas implementadas recentemente para a identificação de expressões faciais, podemos destacar o trabalho de (CARCAGNI; DISTANTE, 2015) que valendo-se da alta sensibilidade a variações de textura oferecida pelo HOG - *Histogram of Oriented Gradients*, aplica a técnica em imagens de faces do banco de dados Cohn-Kanade (CK+), com o objetivo de realizar a classificação da expressão facial. O extrator de características HOG é formado por histogramas de orientação do gradiente de janelas de pixels que concatenados formam o vetor descritor HOG. A principal vantagem da técnica está na capacidade de destacar contornos devido a mudança de orientação do gradiente em regiões de alta derivada. Contudo, variações na face devido a pose e rotação reduzem a eficiência do método. Para minimizar o problema de rotação, (CARCAGNI; DISTANTE, 2015) realiza um pré-processamento na imagem rotacionando-a.

Outra técnica muito utilizada atualmente em problemas de classificação é a Rede Neural Convolutacional (CNN), também conhecida como *Deep Learning*. Em (ZHOU; SHI, 2016), demonstrou-se a partir dos bancos de imagens Cohn-Kanade (CK+) e Toronto Face Dataset (TFD), a grande capacidade do método aplicado no reconhecimento de expressões faciais, às custas de um grande custo computacional. A rede foi analisada qualitativamente, visualizando os padrões de textura que excitavam os diferentes neurônios nas camadas convolucionais demonstrando que a CNN treinada se assemelhava a percepção humana. O descritor utilizado como entrada da rede foi os pixels da imagem. De forma semelhante ao trabalho (CARCAGNI; DISTANTE, 2015), as imagens de entrada necessitam ser pré-processadas de modo a apresentarem pose semelhante as imagens submetidas à etapa de treinamento.

Outro extrator de característica empregado para expressões faciais é o *Local Binary Patterns* (LBP), (PIRES; NETO, 2015) emprega o método para avaliar as informações de texturas em determinadas regiões da face, neste método a intensidade de cada pixel é substituída por um valor binário, determinado pela comparação entre a intensidade do pixel e as intensidades dos pixels vizinhos. Os descritores de textura são extraídos calculando-se o histograma de intensidade de pixels de cada região isoladamente. Posteriormente, os histogramas de cada região são concatenados constituindo o vetor de características. De maneira similar, em (JABID; CHAE, 2010) aplicou-se a técnica *Local Directional Pattern* (LDP), entretanto, ao invés de avaliar a intensidade dos pixel de uma região, esse trabalho considera a magnitude dos valores das bordas em diferentes direções com o propósito de descrever os atributos necessários para a etapa de classificação. Ambas as abordagens baseadas em LBP utilizaram o banco de imagens Cohn-Kanade (CK+), contudo (JABID; CHAE, 2010) alcançou taxas de classificação superiores a (PIRES; NETO, 2015).

Em (KUNG; BOUCHAFFRA, 2015), foi proposto a aplicação do método 3D-HMM (Modelo Oculto de Markov 3D) para o reconhecimento de expressões faciais e suas emoção associadas em evolução temporal. Um Histograma de Fluxo Óptico é usado como descritor de características, enquanto o treinamento e os testes são realizados através 3D-HMM, sendo que a nomenclatura 3D se refere a ideia de volume gerada a partir da concatenação dos histogramas (2D) no tempo, a terceira dimensão. A partir de uma sequência de *frames* de uma face, a técnica trata as variações de histograma como um processo estocástico, gerando assim um modelo estatístico para a evolução das expressões. O modelo é então usado para a etapa classificação da expressão facial interpretando uma nova sequência de eventos, a partir do Banco de dados MMI Facial Expression Database (PANTIC et al., 2005) atingindo acurácia de 90%.

Em (PEDROSO; SALLES, 2012), utilizou-se a modelagem da face por meio do algoritmo AAM - *Active Appearance Model* com objetivo de encontrar uma máscara de pontos que se conforme adequadamente em uma face, ainda que a mesma tenha deformações criadas por pose ou devido a expressões faciais. É importante destacar que ao contrário dos trabalhos supracitados, o sistema proposto por (PEDROSO; SALLES, 2012) não requer nenhum pré-processamento nas imagens de teste, o que configura vantagem. O vetor descritor empregado é constituído pela posição x e y de cada ponto da máscara de ajuste AAM. O banco de dados utilizados para treinamento e validação do modelo em (PEDROSO; SALLES, 2012) foi o *The Japanese Female Facial Expression Database - JAFFE*. O sistema apresentou uma taxa de reconhecimento de 55,4%

Em (MARTIN; WERNER; , 2008), assim como em (PEDROSO; SALLES, 2012), foi aplicado o *Active Appearance Model* para extrair características faciais com o objetivo de reconhecer expressões faciais. Na etapa de classificação foram utilizados os algoritmos Support Vector Machines e o *Linear Discriminant Analysis* (LDA) combinado com a

distancia de Mahalanobis. O sistema apresentou a melhor taxa de reconhecimento para o classificador SVM com acurácia de 80%. Nesse mesmo trabalho, foi aplicado o AAM para estimar a pose 6DOF da cabeça a partir de uma câmera monocular.

1.5 Organização do Texto

Neste capítulo foi definido o problema a ser tratado, a motivação, os objetivos do trabalho, a metodologia proposta, bem como o estado da arte sobre o tema. No capítulo 2 é definida a ferramenta *Active Appearance Model* - AAM necessária para a modelagem da face. No capítulo 3 são descritos os algoritmos escolhidos para o processo de classificação. Já no capítulo 4 a metodologia adotada na pesquisa é então aplicada. Inicialmente, apresenta-se os bancos de dados utilizados e posteriormente o desenvolvimento do método proposto. Já no capítulo 5 são apresentados os resultados obtidos, bem como a análise dos mesmos. No capítulo 6, finalmente são expostas as conclusões alcançadas e então sugere possíveis melhorias para o sistema proposto, assim como outras aplicações para ferramenta AAM. No apêndice A é descrito o PCA - Análise de Componente Principais, instrumento essencial para a construção do algoritmo AAM.

2 Active Appearance Model - AAM

2.1 Definição

O *Active Appearance Model* (AAM) é um algoritmo de visão computacional que busca ajustar uma máscara de pontos generalista à uma classe de objeto, a construção AAM - *Active Appearance Model* é estruturada com base em um modelo de forma (*Shape Model*) e um modelo de textura (*Appearance Model*) que combinados buscam descrever estatisticamente uma classe de objeto. O ajuste do AAM é realizado por meio de um processo de minimização do erro entre o modelo de treinamento e um determinado objeto parametrizado (MATTHEWS; BAKER, 2004).

2.2 Construção do Modelo

Fundamentado em um conjunto de imagens que proporcione a melhor representatividade possível, tanto em relação a aparência quanto suas possíveis deformações. A composição do modelo de forma e textura requer um conjunto n de k marcações (*landmarks*), $[x_1, y_1, \dots, x_k, y_k]$ para cada imagem do conjunto de treinamento. Esses pontos definem os principais contornos do objeto na imagem, como demonstrado na Figura 6. Seguindo como referência (LUCEY et al., 2010), foi escolhido o número de 68 *landmarks* e adotado o mesmo padrão de posição.



Figura 6 – Marcações (*Landmarks.*) Fonte: Modificada a partir do banco de dados CK+.

2.2.1 Construção do Modelo de Forma

2.2.1.1 Procrustes Analysis

Como observado na Figura 7, a sobreposição das marcações de todas as imagens em um plano gera uma dispersão de pontos que os tornam inviáveis para a construção do modelo de forma, tendo em vista que o resultado obtido não apresenta nenhum padrão de dispersão. Isto ocorre devido às transformações de similaridades (posição, escala e orientação) entre as instâncias contidas nas diversas imagens de treinamento. Por esta razão, se faz necessário a uso do método *Procrustes Analysis*¹ que tem por finalidade minimizar as transformações de similaridade. O algoritmo consiste na translação, rotação e/ou escalonamento das marcações de cada instância.

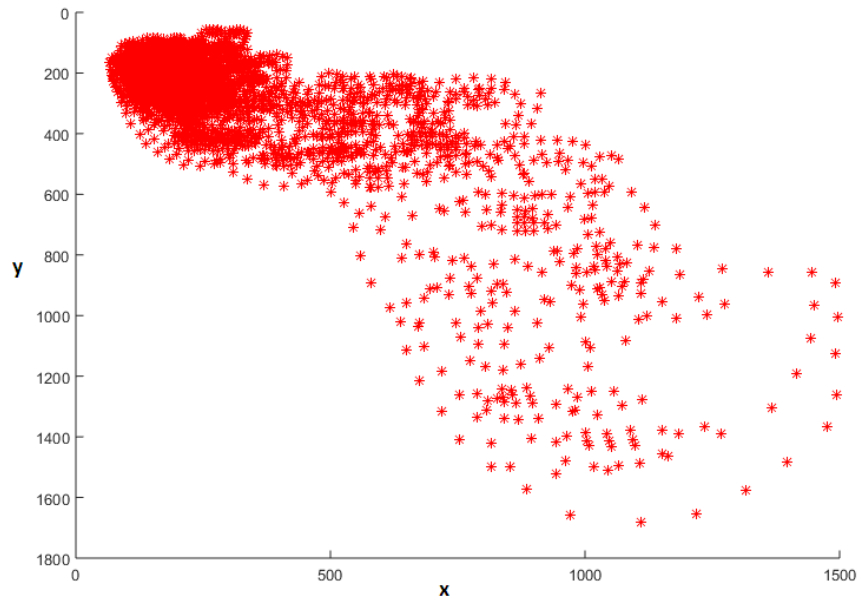


Figura 7 – *Procrustes Analysis* - *Dados de entrada*. Fonte: Própria.

A componente deslocamento é removida transladando a máscara de pontos de modo que o centróide de cada instância fique localizado na origem. Matematicamente, dados k pontos de 2 dimensões,

$$[(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)], \quad (2.1)$$

o centroide (\bar{x}, \bar{y}) pode ser obtido da seguinte forma:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_k}{k}, \quad (2.2)$$

$$\bar{y} = \frac{y_1 + y_2 + \dots + y_k}{k}. \quad (2.3)$$

¹ O nome Procrustes refere-se a um criminoso da mitologia grega que fez suas vítimas caberem em sua cama, esticando seus membros ou cortando-os.

Então a translação de cada ponto se dá fazendo,

$$[(x_1 - \bar{x}, y_1 - \bar{y}), (x_2 - \bar{x}, y_2 - \bar{y}), \dots, (x_k - \bar{x}, y_k - \bar{y})] \quad (2.4)$$

Da mesma forma, a componente escalar é removida dividindo cada ponto pelo fator

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2 + \dots + (x_k - \bar{x})^2 + (y_k - \bar{y})^2}{k}}, \quad (2.5)$$

de modo que o valor da raiz quadrada da média das distâncias (RMSD) em relação a origem seja 1. Assim, gera-se os pontos:

$$[((x_1 - \bar{x})/s, (y_1 - \bar{y})/s), ((x_2 - \bar{x})/s, (y_2 - \bar{y})/s), \dots, ((x_k - \bar{x})/s, (y_k - \bar{y})/s)]. \quad (2.6)$$

A remoção da componente rotacional é mais complexa, uma vez que a orientação de referência padrão nem sempre está disponível. Todavia, devido ao fato do objeto em questão está, em geral, alinhado verticalmente, o objeto de referência pode ser definido pela média de todas máscaras de pontos.

Uma vez fixado o objeto de referência, todas as instâncias são rotacionadas em relação a origem de forma que seja encontrado um ângulo que minimize o quadrado da distância entre os pontos correlatos.

A Figura 8 exhibe o resultado da análise procrustes. Por meio dela, nota-se que a nuvem de pontos forma uma dispersão uniforme que se assemelha à face humana, em contraposição aos dados não tratados mostrados na Figura 8.

2.2.1.2 Modelo de Forma - (*Shape Model*)

Com as n instâncias γ normalizadas, a primeira etapa para se obter o modelo de forma, (*shape model*), é concatenar cada vetor de *landmarks*

$$\gamma_n = [x_1, y_1, x_2, y_2, \dots, x_k, y_k], \quad (2.7)$$

em uma matriz

$$M = [\gamma_1^T, \gamma_2^T, \dots, \gamma_n^T], \quad (2.8)$$

onde T é o operado transposto.

A Figura 9 fornece a representação da forma média S_0 das n instâncias calculadas a partir da média das linhas da matriz M .

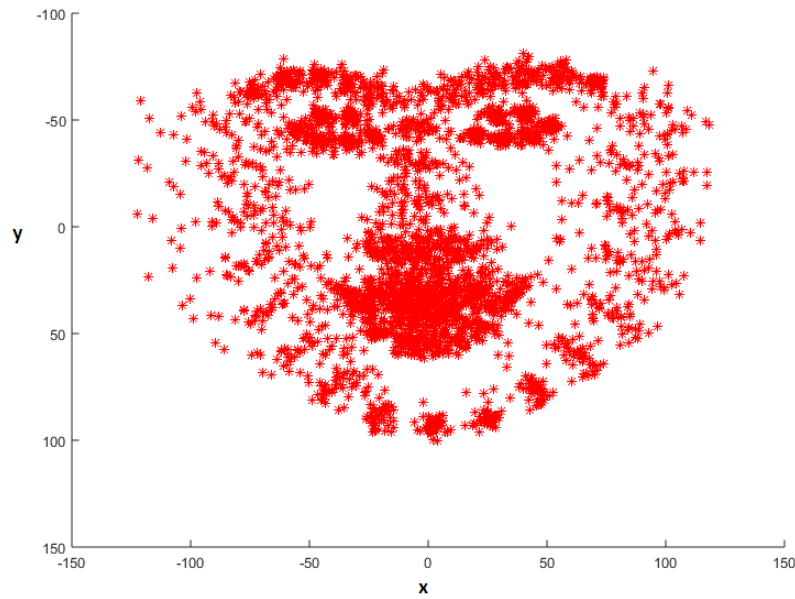


Figura 8 – *Procrustes Analysis* - Dados tratados. Fonte: Própria.

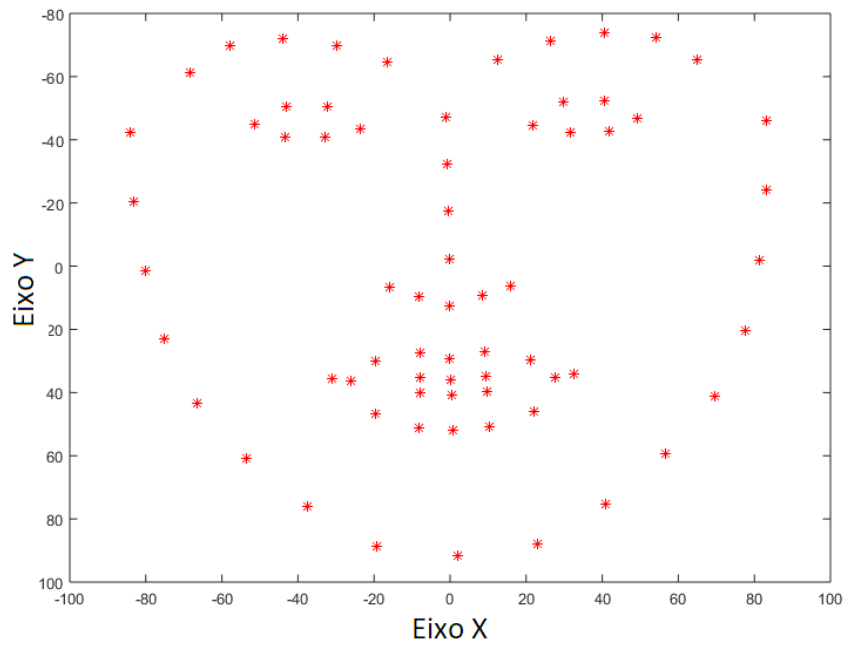


Figura 9 – *Procrustes Analysis* - Forma média. Fonte: Própria.

Devido à capacidade de seleção de características representativas, bem como na compressão dos dados, a Análise de Componentes Principais (PCA), discutida no Apêndice A, aplicada em (2.8) e juntamente com a forma média, obtendo assim o modelo de forma \hat{S} (*Shape Model*) definido pela equação:

$$\hat{S} = S_0 + \sum_{i=1}^n S_i p_i, \quad (2.9)$$

onde S_0 representa a forma média, S_i os i maiores autovetores da matriz de

dispersão amostral obtida a partir da matriz M e p_i os parâmetros do modelo de forma a serem estimados. Para esse trabalho, foram utilizados os 14 autovetores de maior energia ortonormalizados de modo que 95% da representatividade dos dados fosse mantida, a Figura 10 apresenta a distribuição de energia dos autovetores.

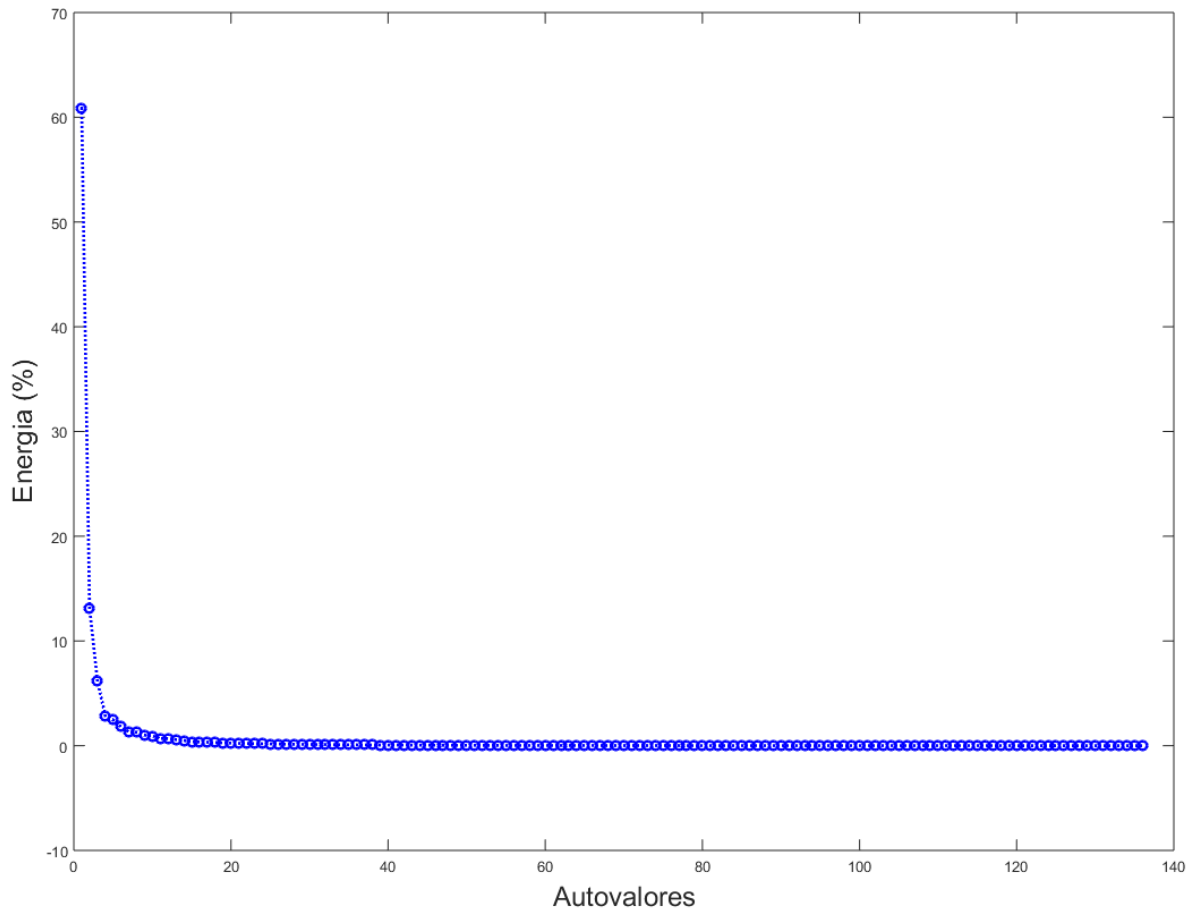


Figura 10 – Energia dos autovetores. Fonte: Própria.

As etapas necessárias para implementação da abordagem proposta para a construção do modelo de forma é apresentado na Figura 11.

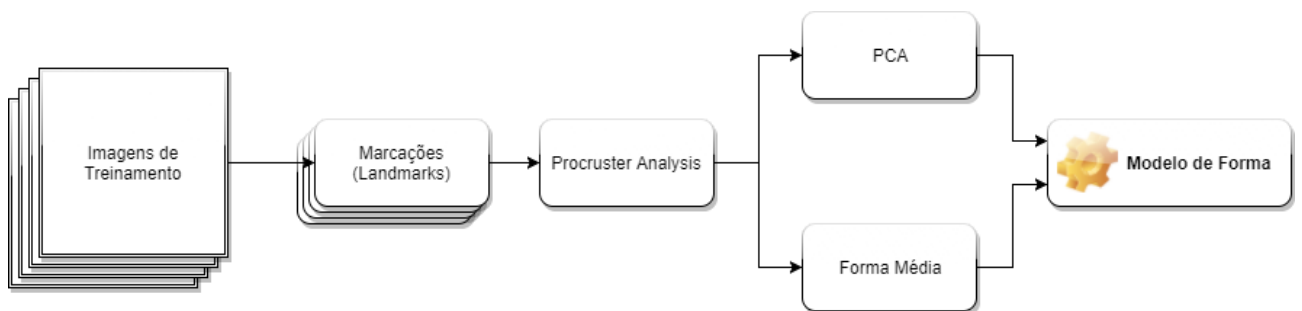


Figura 11 – Etapas para a construção do modelo de forma. Fonte: Própria.

2.2.2 Construção do Modelo de Textura

Para a construção do modelo de textura é necessário remover de todas as imagens de treinamento as deformações geradas por escala e pose, visto que inicialmente cada imagem possui o objeto disposto na cena de diferentes formas. De modo a solucionar este problema, é necessário realizar um mapeamento de pixels de forma a levá-los de uma posição indesejada, gerada pelas deformações, para uma posição pretendida. Com a finalidade de executar esse processo de mapeamento é preciso implementar duas etapas: a triangulação de Delaunay entre as *landmarks* e função de deformação *Piecewise Affine Warps*, definida como $\mathbf{W}(\mathbf{x}; \mathbf{p})$, sendo que representa um pixel com posição (x, y) e \mathbf{p} um vetor de parâmetros que mapeia \mathbf{x} em uma nova posição (x, y) .

2.2.2.1 Triangulação de Delaunay

A primeira etapa para construção do modelo de textura é a demarcação de uma região que delimitará pixels pertencentes a essa região. Utilizando as *landmarks* concomitante com a triangulação de Delaunay busca-se obter as delimitações de pixels necessárias.

Um conjunto de pontos em um plano euclidiano terá inúmeras possibilidades de triangulação. É natural questionar se existe alguma configuração que melhor represente as características de textura. Em geral, a geração de triângulos com base estreita, quando comparado com a sua altura, é indesejável devido a pobreza de informação. A triangulação de Delaunay é um dos algoritmos mais importantes da Geometria Computacional, uma vez que, dado um conjunto de pontos, obtêm-se uma triangulação cujo o valor do menor ângulo de cada triângulo seja maximizado, gerando de forma única, a melhor configuração de triângulos para descrever a textura de uma superfície.

Para o conjunto de vértices da Figura 12(a), é definido uma triangulação, Figura 12(b), em seguida são inseridos círculos inscritos sobre os vértices de todos os triângulos, Figura 12(c). A condição para um triângulo ser de Delaunay é não haver nenhum outro vértice no interior do círculo.

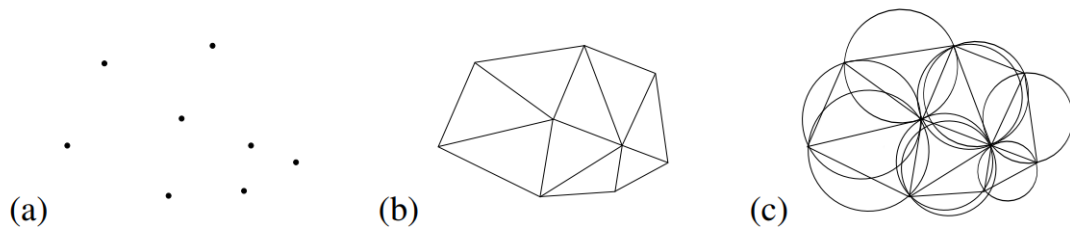


Figura 12 – Triangulação de Delaunay. Fonte: Própria.

Caso haja um vértice no interior do círculo isso indicará uma aresta indesejada. Desse modo, um método utilizado nos algoritmos de Delaunay para transformá-la em uma aresta válida é a troca (*flip*) dessa aresta. Observe, na Figura 13 que, pra a realização da

troca de arestas, avalia-se os ângulos $b\hat{e}a$ e $b\hat{c}a$ na hipótese da soma dos ângulos tenha valor maior que 180° a aresta \bar{ab} é trocada por \bar{ce} .

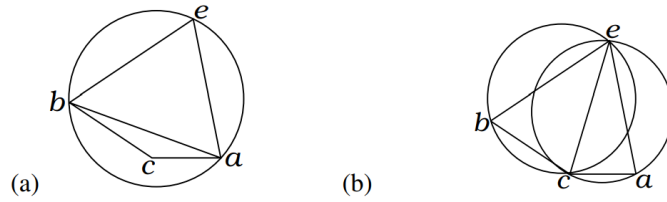


Figura 13 – Triangulação de Delaunay - *Flip*. Fonte: Própria.

A Figura 13(a), exemplifica uma aresta ilegal gerada por um vértice c , já a Figura 13(b), ilustra a troca de aresta e a maximização dos ângulos mínimos.

2.2.2.2 Piecewise Affine Warp

Portanto, com a triangulação de Delaunay pode-se realizar uma correspondência de triângulos da imagem de entrada com a forma média. Sendo assim, é possível mapear todos pixels das imagens de treinamento para uma posição desejada, atenuando as variações geradas pelas transformações de similaridades entre as imagens treinamento. A representação do processo de triangulação de Delaunay e o mapeamento de pixels é apresentada na Figura 14, onde o triângulo, preenchido pela cor vermelha, da face esquerda possui um correspondente na face da direita. Como pode ser visto, esse processo removeu as deformações do objeto face, geradas por escala e pose, levando a face de uma pose levemente rotacionada para uma pose frontal. Também é preciso destacar que o processo de mapeamento não é perfeito, principalmente devido a problemas de interpolação de pixels, provocando um deterioramento da textura em algumas regiões, prejudicando a qualidade da imagem.

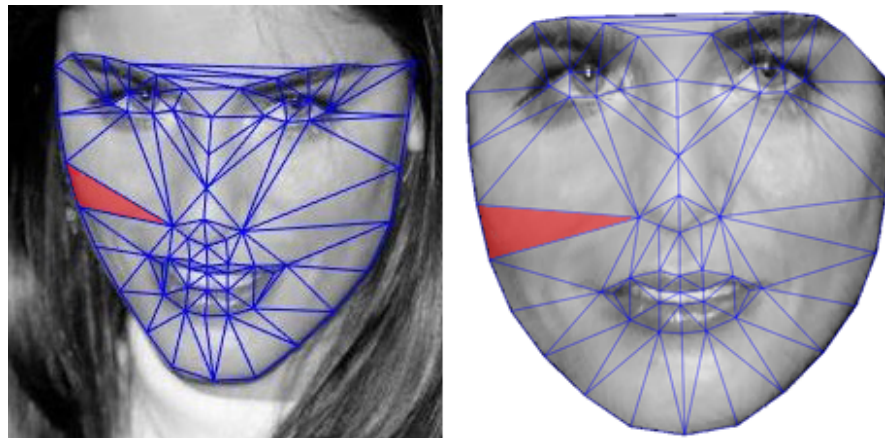


Figura 14 – Triangulação de Delaunay. Fonte: Modificada a partir do banco de dados CK+.

Para se realizar o mapeamento, o algoritmo AAM, se utiliza a função *Piecewise Affine Warps* definida por $\mathbf{W}(\mathbf{x}; \mathbf{p})$ onde $\mathbf{p} = (p_1, \dots, p_n)^T$ é o vetor de parâmetros capaz de mapear o pixel \mathbf{x} para as coordenadas do modelo canônico.

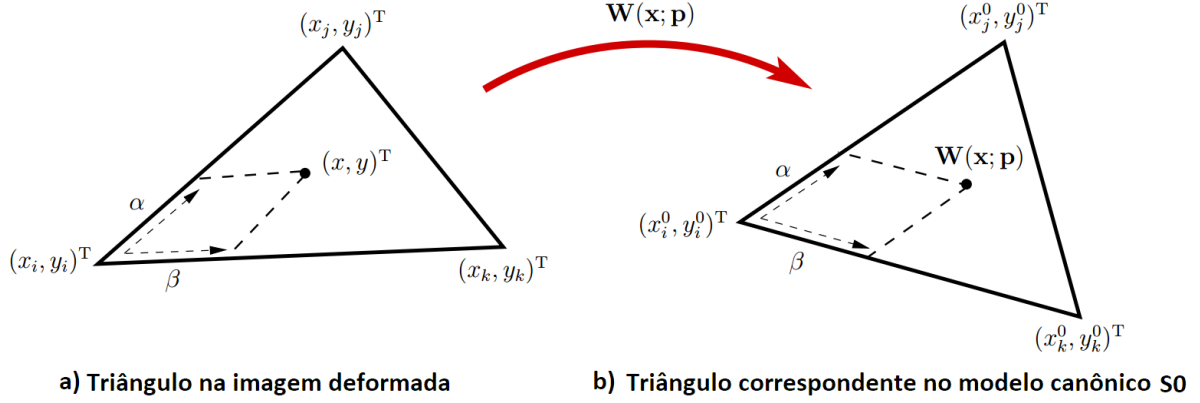


Figura 15 – Função de Mapeamento - *Piecewise Affine Warp*. Fonte: (MATTHEWS; BAKER, 2004).

Para cada pixel \mathbf{x} contido no triângulo S_n com vértices $(x_i^n, y_i^n)^T$, $(x_j^n, y_j^n)^T$, $(x_k^n, y_k^n)^T$ da imagem de entrada obtidos pela triangulação de Delaunay é calculado um $\mathbf{W}(\mathbf{x}; \mathbf{p})$ mapeando cada pixel \mathbf{x} para o triângulo correspondente S_0 com vértices $(x_i^0, y_i^0)^T$, $(x_j^0, y_j^0)^T$, $(x_k^0, y_k^0)^T$ exemplificado na Figura 15. A esse processo de deformação da imagem I denotaremos a simbologia $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$.

Considerando o pixel $\mathbf{x} = (x, y)^T$ no triângulo S_n com vértices $(x_i^n, y_i^n)^T$, $(x_j^n, y_j^n)^T$, $(x_k^n, y_k^n)^T$. Esse pixel pode ser definido como:

$$\mathbf{x} = (x, y)^T = (x_i^n, y_i^n)^T + \alpha [(x_j^n, y_j^n)^T - (x_i^n, y_i^n)^T] + \beta [(x_k^n, y_k^n)^T - (x_i^n, y_i^n)^T] \quad (2.10)$$

onde

$$\alpha = \frac{(x - x_i^n)(y_k^n - y_i^n) - (y - y_i^n)(x_k^n - x_i^n)}{(x_j^n - x_i^n)(y_k^n - y_i^n) - (y_j^n - y_i^n)(x_k^n - x_i^n)} \quad (2.11)$$

e

$$\beta = \frac{(y - y_i^n)(x_j^n - x_i^n) - (x - x_i^n)(y_j^n - y_i^n)}{(x_j^n - x_i^n)(y_k^n - y_i^n) - (y_j^n - y_i^n)(x_k^n - x_i^n)} \quad (2.12)$$

A projeção do pixel \mathbf{x} de S_n em S_0 se dá utilizando α e β calculados a partir dos vértices de S_n . Desta forma, a função *Piecewise Affine Warp* de mapeamento do pixel \mathbf{x} é então:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = (x_i^0, y_i^0)^T + \alpha [(x_j^0, y_j^0)^T - (x_i^0, y_i^0)^T] + \beta [(x_k^0, y_k^0)^T - (x_i^0, y_i^0)^T] \quad (2.13)$$

onde $(x_i^0, y_i^0)^T$, $(x_j^0, y_j^0)^T$, e $(x_k^0, y_k^0)^T$ são os vértices do triângulo correspondente S_0 . Substituindo-se 2.11 e 2.12 em 2.13 e, a partir dos valores dos vértices dos triângulos S_0 e S_n obtêm-se uma simples deformação afim:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = (a_1 + a_2.x + a_3.y, a_4 + a_5.x + a_6.y)^T \quad (2.14)$$

A título de exemplo, a função que mapeia um pixel \mathbf{x} com coordenadas (x, y) de um triângulo com vértices $((0, 0), (10, 10), (10, 0))$ em um triângulo com vértices $((1, 1), (20, 15), (10, 5))$ calculada a partir de 2.13 é:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = (1 + 0, 9.x + y, 1 + 0, 4.x + y)^T. \quad (2.15)$$

Como demonstrado, os 6 parâmetros da função $\mathbf{W}(\mathbf{x}; \mathbf{p})$ ($a_1, a_2, a_3, a_4, a_5, a_6$) podem ser calculados a partir dos vértices dos triângulos correspondentes. No entanto como consequência da Equação 2.9, também é possível encontrar os parâmetros ($a_1, a_2, a_3, a_4, a_5, a_6$) a partir do vetor p do modelo de forma parametrizado para uma instância. A implementação do processo de deformação $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ pode ser estruturada da seguinte forma:

- Dado o vetor de parâmetros p do modelo de forma, encontrar os vértices $(x^n, y^n)^T$ de todos os triângulos e seus correspondentes vértices nos triângulos canônicos da forma média.
- Calcular $(a_1, a_2, a_3, a_4, a_5, a_6)$ para cada triângulo.
- Finalmente, utilizando a função $\mathbf{W}(\mathbf{x}; \mathbf{p})$, realizar o mapeamento cada pixel contido no triângulo S_n utilizando os correspondentes valores de $(a_1, a_2, a_3, a_4, a_5, a_6)$.

2.2.2.3 Modelo de Textura - (*Appearance Model*)

Definindo a utilização de imagem em tons de cinza, um processo de normalização é realizado para cada imagem do conjunto de treinamento. Com as n instâncias I normalizadas, o modelo de textura (*Appearance Model*) pode ser obtido concatenando-se o vetor de pixels \mathbf{x} de forma lexicográfica das n imagens de treinamento I_n :

$$I_n = [\mathbf{x}_1, \mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \dots, \mathbf{x}_n], \quad (2.16)$$

em uma matriz de pixels

$$P = [I_1^T, I_2^T, \dots, I_n^T]; \quad (2.17)$$

A média das linhas da matriz P fornece a textura média A_0 , Figura 16, que fará parte do modelo de textura.



Figura 16 – Textura média. Fonte: Própria.

Finalmente, como realizado no conjunto de dados do na construção do modelo de forma, o PCA é aplicado no conjunto de texturas para obter-se o modelo da textura, definido por

$$\hat{A} = A_0 + \sum_{j=1}^m A_j c_j, \quad (2.18)$$

sendo A_0 a textura média, que pode ser compreendida como a média dos correspondentes pixels de todas as imagens $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ e A_j os m maiores autovetores ortonormalizados do conjunto de texturas e c_j os parâmetros do modelo a serem estimados. Foram utilizados os 200 autovetores de maior energia ortonormalizados de modo que 97% da representatividade dos dados fosse mantida. A Figura 17 apresenta a distribuição de energia dos autovalores.

Um esquemático das etapas necessárias para a construção do modelo de textura é apresentado na Figura 18.

2.2.3 Modelo Completo e Ajuste AAM

O modelo completo do AAM dá-se da junção do modelo de forma e de textura, Equações 2.9 e 2.18 respectivamente, visando descrever a classe de objeto a ser modelado com informações de formato e textura. A Figura 19 apresenta de forma intuitiva a o algoritmo de ajuste AAM com seu modelo completo.

Sendo assim, a partir de uma nova imagem I de entrada, o ajuste automático do AAM implica em estimar os parâmetros p e c das Equações 2.9, 2.18, respectivamente de tal que a norma l_2 do erro entre o modelo da textura e a imagem de entrada deformada pela função $\mathbf{W}(\mathbf{x}; \mathbf{p})$ seja minimizado, sendo que m os 200 maiores autovetores ortonormalizados do conjunto de texturas. O processo de minimização de

$$\operatorname{argmin}_{p,c} \| I(W(x;p)) - A_0 - \sum_{j=1}^m A_j c_j \|^2, \quad (2.19)$$

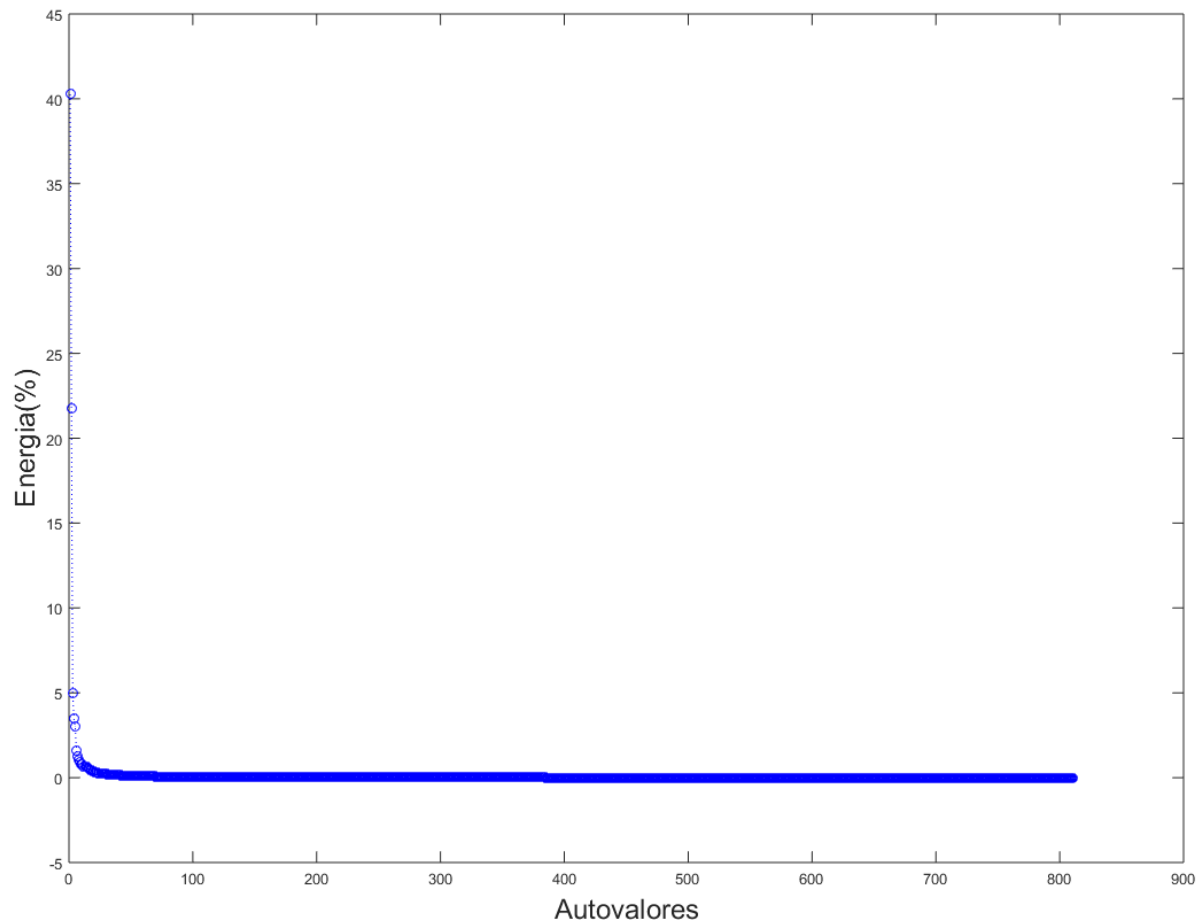


Figura 17 – Energia dos autovetores - Conjunto de Textura. Fonte: Própria.

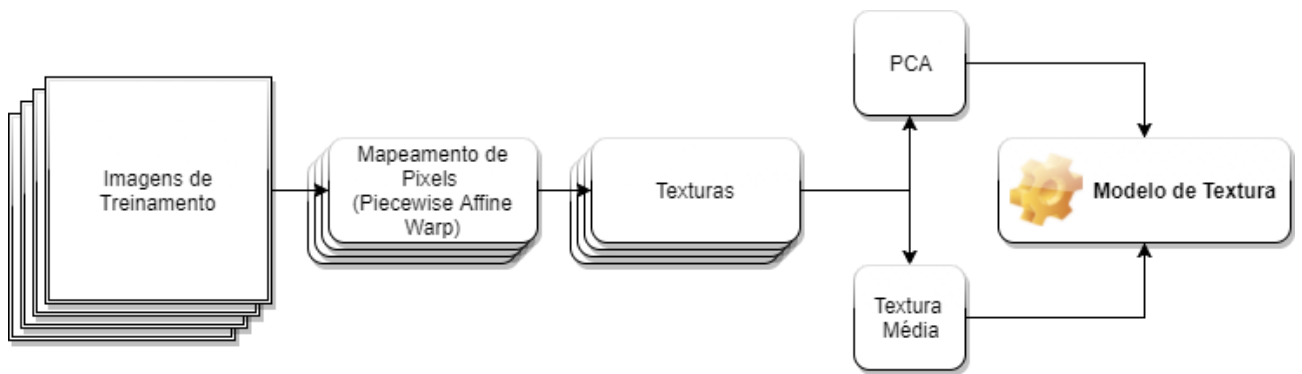


Figura 18 – Etapas para a construção do modelo de textura. Fonte: Própria.

inicializa ambos os parâmetros $p = (0, 0, \dots, 0)$ e $c = (0, 0, \dots, 0)$, e iterativamente os valores são atualizados. Tipicamente, o teste de convergência decorre da verificação da norma l_2 em relação a um limiar, (MATTHEWS; BAKER, 2004).

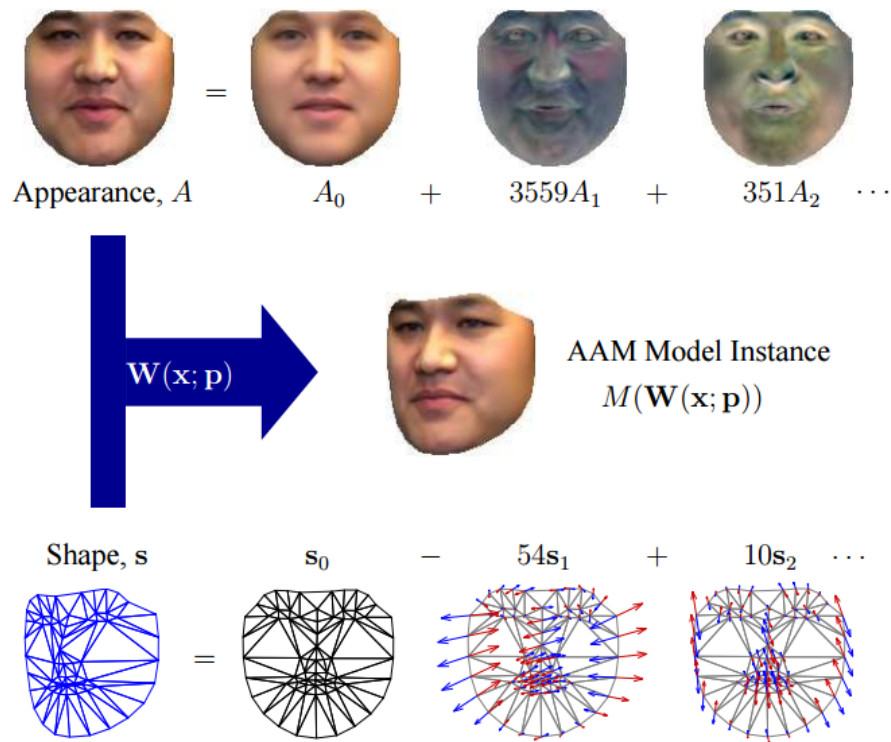


Figura 19 – Algoritmo AAM. Fonte: (MATTHEWS; BAKER, 2004).

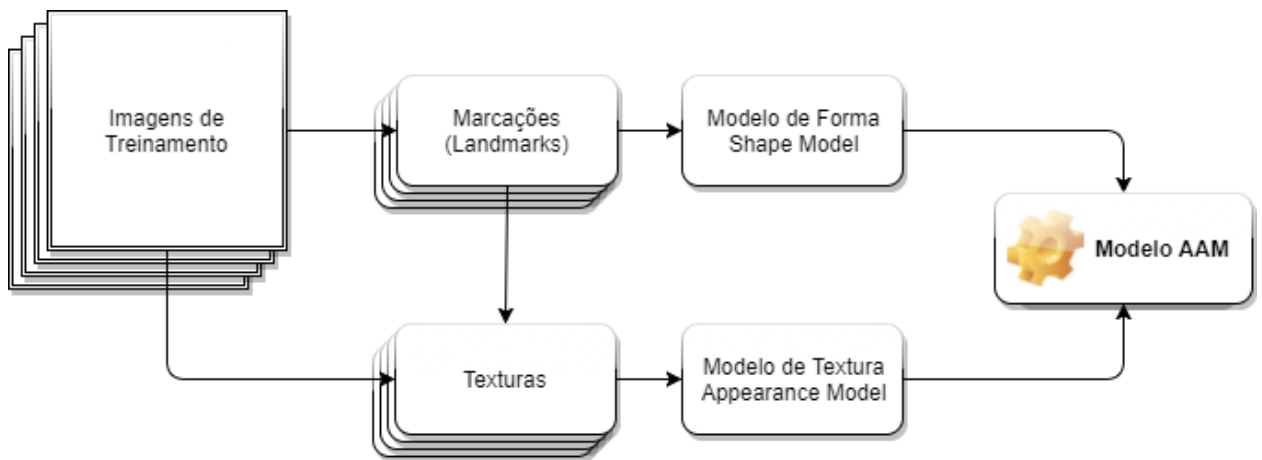


Figura 20 – Diagrama de blocos descrevendo a metodologia de criação do modelo AAM. Fonte: Própria.

2.3 Solução AAM

O AAM é formulado como um problema de fluxo ótico, também conhecido por Lucas-Kanade (BAKER; MATTHEWS, 2003), o qual pode ser resolvido iterativamente usando o método clássico Gauss-Newton. Este método, quando aplicado à solução do AAM se mostra ineficiente. Entretanto, existem outras soluções propostas, dentre elas o método *Inverse Compositional* (IC), variações para esse algoritmo voltadas especificamente para o AAM foram propostas em (BAKER; MATTHEWS, 2003).

Em (MATTHEWS; BAKER, 2004) é apresentado o algoritmo para otimizar a

solução da equação 2.19, *Project Out Inverse Compositional* (POIC), que possui uma execução mais rápida, quando comparado a solução clássica (IC), devido ao seu pré-cálculo das matrizes jacobiana e hessiana.

Gross em (GROSS; BAKER, 2005) introduziu o *Simultaneous Inverse Compositional* (SIC) atualizando simultaneamente os parâmetros de deformação e textura. Tzimiropoulos em (TZIMIROPOULOS; PANTIC, 2013) apresentou os algoritmos *Fast-SIC* e o *Fast-forward* para resolver de forma mais eficientemente o problema do ajuste AAM, com um custo computacional menor que os trabalhos citados anteriormente. O algoritmo *Fast-SIC* apresentou desempenho superior ao *Fast-forward*.

2.3.1 Fast-SIC

No algoritmo *Inverse Compositional* (IC) o modelo da textura é linearizado em relação a $p = 0$, sendo assim, $\mathbf{W}(\mathbf{x}; \mathbf{p})$ se torna a identidade. Uma atualização Δp é então encontrada usando mínimos quadrados e p é atualizado realizando uma composição entre p e Δp , $p \leftarrow p \circ \Delta p$, obtendo assim o novo valor do vetor p . O símbolo \circ se refere ao processo de composição incremental das duas variáveis. Mais detalhes da solução podem ser encontrado em (MATTHEWS; BAKER, 2004).

Em cada iteração do método *Simultaneous Inverse Compositional* (SIC) a equação 2.19 é linearizada em relação a c e p iguais a zero. Isto é equivalente a resolver, a cada iteração, o seguinte problema de optimização, sendo os termos de segunda ordem foram omitidos:

$$\underset{\Delta p, \Delta c}{\operatorname{argmin}} \| I - A_0 - Ac - A\Delta c - J\Delta p \|^2, \quad (2.20)$$

onde Δp e Δc se referem a atualização dos parâmetro do modelo de forma e de textura a serem estimados, já a variável J é definida por:

$$J = J_0 + \sum_{i=1}^m c_i J_i, \quad (2.21)$$

onde

$$J_i = \begin{bmatrix} A_{i,x} & A_{i,y} \end{bmatrix} \frac{\partial(\mathbf{W}(\mathbf{x}; \mathbf{p}))}{\partial(p)}, \quad (2.22)$$

$A_{i,x}$, $A_{i,y}$ o gradiente de A_i em relação a x e y , e $\frac{\partial(\mathbf{W}(\mathbf{x}; \mathbf{p}))}{\partial(p)}$ a matriz jacobiana da função de mapeamento *piecewise affine warp*. A implementação dessa matriz é apresentada em (MATTHEWS; BAKER, 2004).

Em (TZIMIROPOULOS; PANTIC, 2013) é aplicada a teoria de otimização de modo a resolver a equação 2.20 por um caminho mais eficiente. A solução apresentada é então nomeada de algoritmo *Fast-SIC*. A partir da definição

$$\min_{x,y} f(x,y) = \min_x [\min_y f(x,y)], \quad (2.23)$$

a Equação 2.19 é otimizada calculando primeiramente Δc , resultando em

$$\Delta c = A^T(I - A_0 - Ac - J\Delta p). \quad (2.24)$$

Aplicando Δc na equação 2.20 temos a solução otimizada

$$\operatorname{argmin}_{\Delta p} \|I - A_0 - J\Delta p\|^2, \quad (2.25)$$

onde o Δp que a minimiza é definido por

$$\Delta p = H_{fsic}^{-1} J_{fsic}^T (I - A_0), \quad (2.26)$$

onde

$$J_{fsic} = \begin{bmatrix} A_x c' & A_y c' \end{bmatrix} \frac{\partial \mathbf{W}}{\partial p}, \quad (2.27)$$

$$H_{fsic} = J_{fsic}^T J_{fsic}, \quad (2.28)$$

e

$$c' = \begin{bmatrix} 1 \\ c \end{bmatrix}. \quad (2.29)$$

Devido ao fato de J_{fsic} ser em função de c , a Equação 2.27 necessita ser recalculada a cada iteração.

Para exemplificar, Algoritmo 1 mostra como é feito o Ajuste do AAM.

Algoritmo 1: Ajuste AAM

Entrada: Imagem de uma face, *Shape Model*, *Appearance Model*.

Saída: Instância \hat{S} do modelo de forma ajustada à face de entrada.

Início: Pré-posicionamento da máscara de pontos na face.

Inicialização dos parâmetros p e c :

$p = (0, 0, \dots, 0);$

$c = (0, 0, \dots, 0);$

while erro > limiar **do**

Warp da imagem de entrada para a forma média:

$I_w = I(W(x; p));$

Atualização do modelo de textura:

$c = A^{-1} * (I_w - A_0);$

$\hat{A} = A_0 + A * c;$

Construção de J_{fsic} e H_{fsic} :

$[Ax, Ay] = \text{gradiente}(\hat{A});$

$J = \text{jacobiano}(Ax, Ay);$

$J_{fsic} = J - A * (A^T * J);$

$H_{fsic} = J_{fsic}^T * J_{fsic};$

$H_{fsic}^{-1} = \text{inv}(H_{fsic});$

Atualização do parâmetros p e c :

$\Delta p = H_{fsic}^{-1} * J_{fsic}^T * (I_w - A_0);$

$\Delta c = A^T * (I_w - I_{rec} - J * \Delta p);$

$c = c + \Delta c;$

$p = p + \Delta p;$

Atualização do modelo de forma:

$\hat{S} = S_0 + \sum_{i=1}^n S_i p_i;$

Cálculo da norma l_2 :

$\text{erro} = \| I(W(x; p)) - A_0 - \sum_{j=1}^m A_j c_j \|^2;$

end while

FIM;

3 Classificadores

Com base no conjunto de vetores descritores que contém os atributos que caracterizam cada instância das possíveis classes, é necessário rotular cada um desses vetores como uma das expressões básicas definidas por Paul Ekman. Para esse propósito, algoritmos de aprendizado de máquina são empregados com o objetivo de encontrar padrões dentro de um conjunto de dados. O problema em questão será abordado por técnicas de classificação supervisionada. Assim sendo, para essa etapa escolhidos 4 métodos muito difundidos em aprendizado de máquina: *k-Nearest Neighbour* (k-nn), *Support Vector Machine* (SVM), Rede Neural Multicamada e Árvore de Decisão.

3.1 *k-Nearest Neighbour* - (k-nn)

Um dos métodos de classificação mais antigos e simples é o k-vizinhos mais próximos (k-nn, do inglês *k-Nearest Neighbors*). Esse método, apesar de sua simplicidade, tem alcançado bom desempenho em diferentes problemas de classificação. Desse modo, um conjunto de dados D pode ser classificado de acordo com as seguintes etapas:

- A distância entre o dado de entrada d e o conjunto de dados de treinamento é calculada utilizando alguma medida de similaridade entre dados, tal como distância Euclidiana, distância Manhattan, distância Minkowski e a medida do cosseno.
- As k instâncias de treino mais próximas, isto é, mais semelhantes ao dado d são selecionados.
- O dado d é classificado em determinada categoria que apresente o maior número de instâncias similares.

Baseado no conceito de que uma expressão facial pode ser representada por um vetor n -dimensional de características, o classificador k-nn pode ser empregado com o propósito de se determinar a qual categoria de expressão uma dada face pertence. A Figura 21 apresenta uma representação gráfica simplificada desse conceito.

A definição do valor mais adequado para k configura outra decisão de grande relevância para o desempenho do método k-nn. É importante ressaltar que o k-nn, para $k=1$, constitui o que é conhecido como Regra do Vizinho mais Próximo. Nesse caso, apesar de sua simplicidade conceitual e baixo custo computacional, dita regra possui um importante comportamento assintótico: o erro de classificação cometido é maior que o

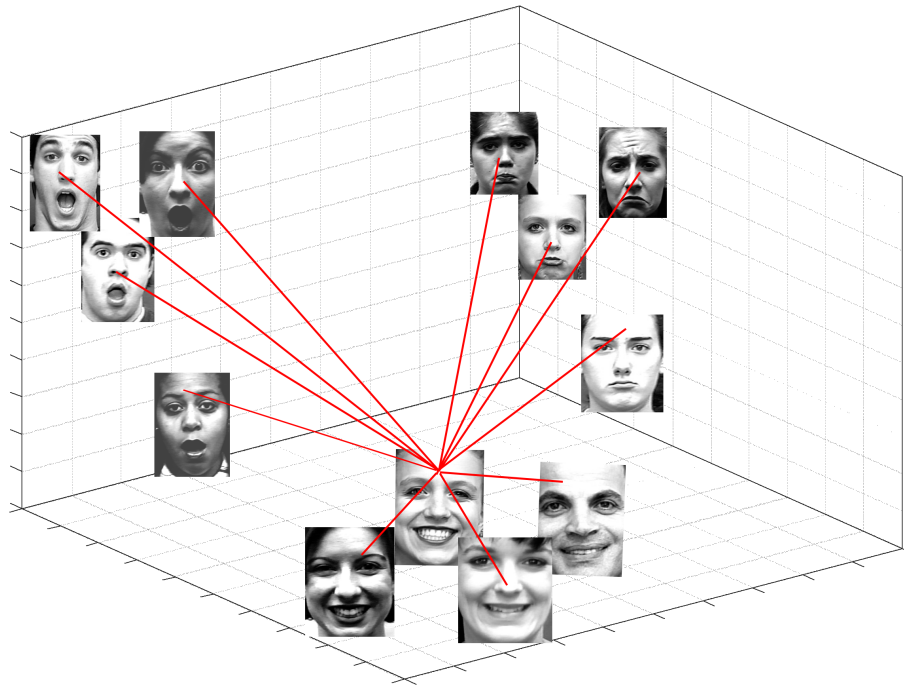


Figura 21 – Representação gráfica simplificada do algoritmo k-nn. Fonte: Modificada a partir do banco de dados CK+.

mínimo possível obtido por um classificador bayesiano, mas nunca pior que duas vezes a taxa de erro de Bayes alcançado (DUDA; HART; STORK, 2000).

3.2 Rede Neural Multicamada

Uma Rede Neural Artificial é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural de armazenar conhecimento experimental e torná-lo disponível para uso (HAYKIN, 2009). Estas unidades, chamadas de neurônios, são dispositivos não-lineares e adaptáveis, embora muito simples em termos de poder computacional e de memória. Porém, quando interligadas, apresentam um enorme potencial para mapeamentos não-lineares. O algoritmo de aprendizagem é o procedimento utilizado para realizar o processo de aprendizagem, cuja função é modificar os pesos sinápticos da rede e de uma forma ordenada para alcançar um objetivo de projeto desejado (HAYKIN, 2009).

A topologia da rede diz respeito ao número de entradas, saídas, número de camadas, número de neurônios por camada e função de ativação. Uma Rede Neural Perceptron Multicamada (MLP) é subdividida nas seguintes camadas: camada de entrada, camada(s) oculta(s) e camada de saída. O funcionamento de uma rede MLP é síncrono, ou seja, dado

um vetor de entrada, este é propagado para a saída multiplicando-se pelos pesos de cada camada, aplicando-se a função de ativação.

A unidade básica de processamento, rotulado de neurônio, cujo modelo é apresentado na Figura 22 é definido matematicamente por meio da seguinte equação:

$$y_k = \varphi(w_k x + b_k), \quad (3.1)$$

sendo o vetor de entrada definido por x , φ a função de ativação que limita a um valor finito o intervalo de amplitude do sinal de saída; w_k o vetor dos pesos referentes a cada entrada do neurônio; b_k é um *offset*, que tem por objetivo realizar um incremento ou decremento ao valor obtido por $w_k x$. Por fim y_k define a saída do neurônio.

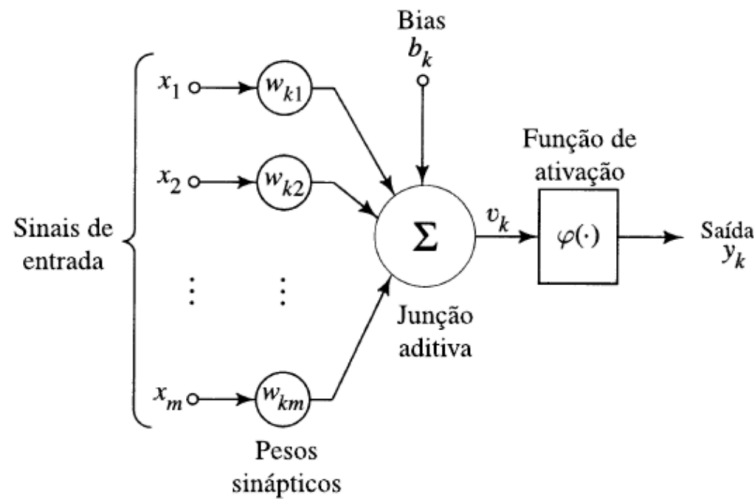


Figura 22 – Modelo de um neurônio. Fonte: (HAYKIN, 2009)

O conjunto de dados destinado à criação do classificador deve ser dividido nos subconjuntos de treinamento, validação e teste. Como usual, o conjunto de treinamento é usado para ajuste de pesos, o conjunto de validação é usado para estimar a capacidade de generalização da rede durante o treinamento, e o conjunto de teste é usado para avaliar o desempenho final da rede.

A função de ativação utilizada tanto para os neurônios ocultos quanto para os de saída foi a tangente sigmóide, sua escolha se deve ao fato de ser o tipo de função de ativação mais utilizado em redes neurais artificiais. É definida como uma função crescente, que apresenta um balanço entre o comportamento linear e não-linear. Um exemplo de função sigmóide é a função tangente hiperbólica, definida pela equação:

$$\varphi(v) = \frac{1 - e^{-av}}{1 + e^{av}} \quad (3.2)$$

O treinamento de uma Rede neural MPL é supervisionado, em outras palavras, o treinamento é realizado a partir dos dados de treinamento classificados com objetivo de

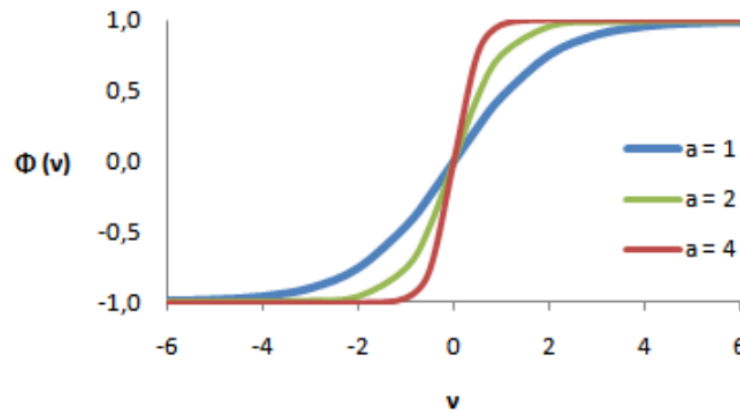


Figura 23 – Função sigmóide com parâmetro de inclinação a variável. Fonte: Própria.

se encontrar os pesos w_k para cada neurônio. A saída da RNA é comparada com a saída desejada, e então o resíduo é usado para ajustar os pesos de acordo com o algoritmo de treinamento. Devido a sua alta eficiência, (HAGAN; MENHAJ, 1994), o algoritmo de Levenberg-Marquardt por retropropagação é utilizado no treinamento da rede utilizada nesse trabalho. A solução envolve uma minimização do erro quadrático médio dos resultados alcançados pela rede e o *Ground truth* dos dados de treinamento.

3.3 SVM - Support Vector Machine

O método de classificação *Support Vector Machines* (SVM), traduzido de forma direta como Máquina de Vetores Suporte é fundamentada na Teoria de Aprendizado Estatístico ou teoria VC (Vapnik-Chervonenkis). O SVM busca a minimização do risco estrutural, o que significa minimizar a probabilidade de classificação errônea de instâncias apresentadas a máquina que não fazem parte dos dados de treinamento (VAPNIK, 1989). As máquinas de vetores-suporte representam uma generalização do algoritmo *Generalized Portrait* desenvolvido na Rússia nos anos oitenta (VAPNIK, 1989). Uma abordagem similar, empregando programação linear ao invés de programação quadrática, foi desenvolvida na mesma época nos Estados Unidos, principalmente por (MANGASARIAN; MUSICANT, 2001). Em sua formulação mais recente, a abordagem SVM foi concebida nos laboratórios da AT&T em (CORTES; VAPNIK, 1995).

3.3.1 Classes Linearmente Separáveis

O processo de construção do SVM busca encontrar um hiperplano ótimo, responsável pela separação de duas classes. Um conjunto de treinamento é dito linearmente separável se for possível separar as diferentes classes por pelo menos um hiperplano (VAPNIK, 1989). A partir desse conceito, o hiperplano obtido deve gerar a maior margem de separação entre as classes, sendo que as margens obtidas devem possuir a mesma largura para que

não haja preferência sobre nenhuma classe. A Figura 24 demonstra esse conceito, de modo que tanto o hiperplano da direção 1 quanto o hiperplano da direção 2 possuem margens simétricas em relação aos respectivos pontos médio que separam as classes em cada direções. Também é possível observar que o hiperplano na direção 2 apresenta uma margem maior, sendo assim, a escolha desse hiperplano é a mais indicada, devido a sua maior capacidade de separação, minimizando o risco estrutural.

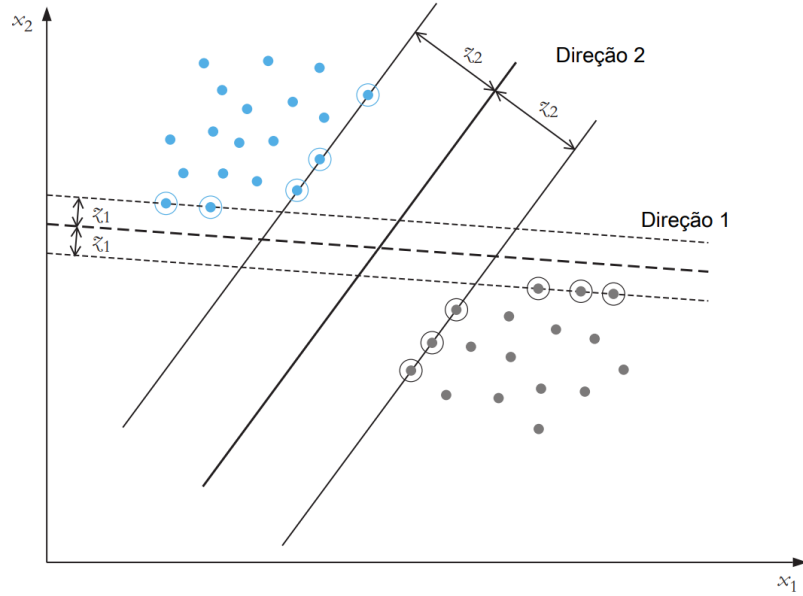


Figura 24 – Hiperplanos separando duas classes a partir de margens iguais. Fonte: (THEODORIDIS; KOUTROUMBAS, 2009).

A equação que gera o hiperplano capaz de separar os padrões é ser definida como o produto escalar entre os vetores \mathbf{w} e \mathbf{x}^T :

$$(\mathbf{w} \cdot \mathbf{x}^T) + b = 0, \quad (3.3)$$

sendo o vetor \mathbf{x}^T a representação dos dados de treinamento, \mathbf{w} o vetor de pesos ajustáveis e b um limiar também conhecido como bias. Portanto o lado caracterizado por $(\mathbf{w} \cdot \mathbf{x}^T) + b > 0$ representa uma classe, e o lado caracterizado por $(\mathbf{w} \cdot \mathbf{x}^T) + b < 0$ representa a outra classe.

Um hiperplano é dito ótimo se ele for capaz de separar o conjunto de vetores \mathbf{x} sem erro e se a distância do hiperplano em relação às classes possuir, de forma simétrica, a maior margem de separação possível, não privilegiando nenhuma classe. Definindo a margem de separação como $\rho(\mathbf{w}, \mathbf{x})$ se obtém a seguinte desigualdade a partir dos vetores de treinamento \mathbf{w}, \mathbf{x} :

$$|\mathbf{w} \cdot \mathbf{x}^T + b| \geq \frac{\rho(\mathbf{w}, \mathbf{x})}{2} \quad (3.4)$$

Fixando a margem de separação em $\rho(\mathbf{w}, \mathbf{x}) = 2$, o problema de maximizar a margem equivale a minimizar a norma de \mathbf{w} , (VAPNIK, 1989). Esse processo resulta então em um hiperplano canônico, onde \mathbf{w} e b devem satisfazer a relação:

$$\underset{\mathbf{w}}{\operatorname{argmin}} |(\mathbf{w} \cdot \mathbf{x}^T + b)| = 1, \quad (3.5)$$

Dessa forma, o hiperplano que separa otimamente os dados é aquele que minimiza

$$\Phi(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2, \quad (3.6)$$

tomando-se um limite arbitrário A para a norma de \mathbf{w} se obtém:

$$\|\mathbf{w}\| \leq A. \quad (3.7)$$

Considerando um ponto $P(x_1, y_1)$ e uma reta r definida por $ax + by + c = 0$. A distância do ponto P a reta r é definida por:

$$d(a, b; P) = \frac{|a.x_1 + b.y_1 + c|}{\sqrt{a^2 + b^2}} \quad (3.8)$$

Logo, a distância $d(\mathbf{w}, b; \mathbf{x})$ de um vetor \mathbf{x} ao hiperplano de separação pode ser expressa na forma:

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|} \quad (3.9)$$

A partir das Equações 3.7 e 3.10 têm-se a relação:

$$d(\mathbf{w}, b; \mathbf{x}) \geq \frac{1}{A} \quad (3.10)$$

Sendo assim, o hiperplano não pode ser mais próximo que $\frac{1}{A}$ para qualquer amostra de treinamento. A Figura 25 mostra, de forma intuitiva, como são reduzidas as possibilidades de hiperplanos, eliminando aqueles que tendem a separar com menor eficácia.

3.3.2 O parâmetro de penalidade C

Um parâmetro que deve ser calibrado na classificação SVM é o valor de $C > 0$, que constitui no parâmetro de custo do algoritmo. Basicamente, o parâmetro C é uma limiar de tolerância aos erros existentes em uma classificação, (VAPNIK, 1989). Dessa forma, o parâmetro determina o equilíbrio entre permitir erros de treinamento e forçar margens rígidas para permitir flexibilidade na separação das classes. Isto cria uma margem suave

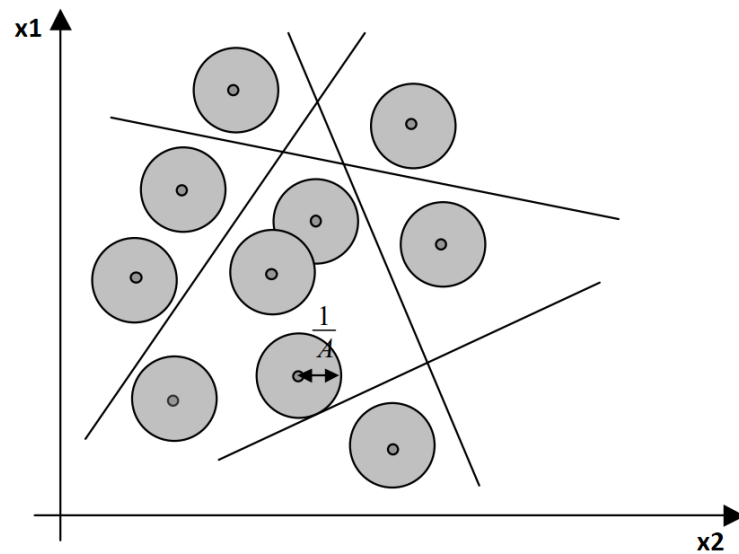


Figura 25 – Os candidatos a hiperplano canônico. Fonte: (THEODORIDIS; KOUTROUMBAS, 2009).

que permite alguns erros de classificação. A Figura 26 ilustra um exemplo de suavização da margem através do parâmetro C por um hiperplano de separação SVM.

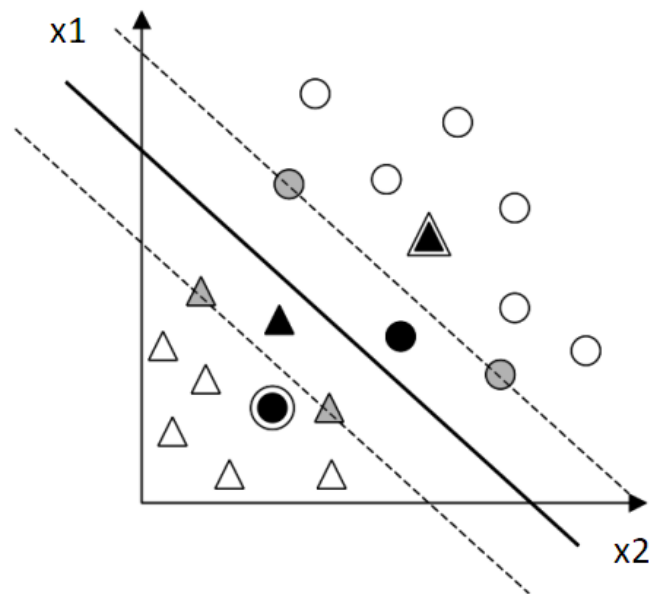


Figura 26 – Parâmetro C de tolerância a erros. Fonte: (THEODORIDIS; KOUTROUMBAS, 2009).

3.3.3 SVM para Multiclasses

Os problemas de classificação podem apresentar um número de classes superior a dois, sendo este tipo de problemas denominado de classificação multi-classe. Na literatura

existem diversas estratégias de decompor os problemas multi-classe em problemas binários. Desta forma, para problemas de N classes, o processo de classificação pode ser subdividido em N problemas de duas classes, sendo que em cada etapa busca-se separar uma classe específica de todas as outras. A essa abordagem é conhecida como um contra todos (THEODORIDIS; KOUTROUMBAS, 2009). A Figura 27 exemplifica o problema de classificação para três classes w_1, w_2 e w_3 , sendo que o hiperplano r_1 é capaz de separar w_1 de w_2 e w_3 , o hiperplano r_2 w_2 de w_1 e w_3 e finalmente, r_3 é o hiperplano que separa w_3 de w_1 e w_2 .

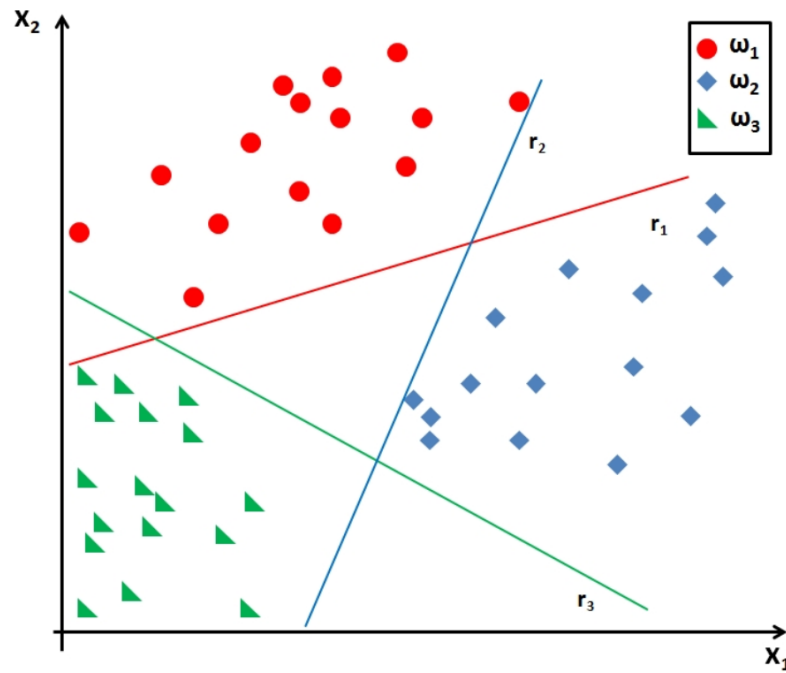


Figura 27 – Classificação SVM multiclasse. Fonte: (PEDROSO; SALLES, 2012).

4 Metodologia

4.1 Bancos de Dados

4.1.1 Cohn-Kanade (CK+) *Database*

Para os experimentos realizados nesse trabalho utilizou-se o banco de dados de expressões faciais Cohn-Kanade (CK+) (KANADE; TIAN, 2000), por ser uma base de dados muito utilizada na literatura, especialmente em trabalhos de reconhecimento facial e de expressões faciais, possibilitando comparações com trabalhos anteriores. O banco é formado por imagens de indivíduos de várias etnias, sexo e idades. As imagens possuem a mesma resolução em pixels, 640 x 480, com todos indivíduos em pose frontal não oclusas, com características de iluminação, *background* padronizadas. A Base CK+ é composta por 10.558 imagens divididas em 123 diretórios principais, sendo cada um corresponde a um indivíduo. Dentro de cada pasta existem subdiretórios contendo imagens referentes à evolução temporal de uma face. Neste caso, inicia-se com uma expressão facial neutra à uma face com expressão (surpresa, medo, tristeza, felicidade, nojo e nervoso) ativada, como se observa na Figura 28. As quantidades de subdiretórios e imagens não são padronizadas por indivíduo. A base CK+ possui, para cada subdiretório, um arquivo de texto com a rotulação da expressão facial correspondente. Além da pré-identificação de cada expressão, o conjunto de imagens do banco possui um arquivo texto contendo as marcações (68 *landmarks*) padrões para o algoritmo AAM, proposto em (LUCEY et al., 2010). Essas informações serão utilizadas como *ground truth* na etapa de treinamento do AAM, bem como na etapa de classificação da expressão facial.

4.1.2 *The Japanese Female Facial Expression (JAFPE) Database*

Outro banco de dados muito utilizado para o problema de reconhecimento de expressões faciais é o JAFPE - *Japanese Female Facial Expression* (LYONS; KAMACHI; GYOBA., 2001). O banco contém um total de 219 imagens de 10 indivíduos femininos de etnia japonesa em pose frontal, como se observa na Figura 29. Cada indivíduo possui imagens para as expressões faciais: surpresa, medo, tristeza, felicidade, nojo e nervoso além da neutra. Alguns indivíduos são apresentados em 3 ou 4 poses para a mesma expressão. As imagens possuem a mesma resolução em pixels, 256 x 256, com todos indivíduos em pose frontal não oclusas, com características de iluminação, *background* padronizadas. Ao contrário da base CK+, o conjunto de imagens do banco de dados JAFPE não possui a marcação (*landmarks*) padrão AAM. No entanto, a informação de *ground truth* da expressão facial está disponível no nome do arquivo de cada imagem. A utilização da



Figura 28 – Sequencia de imagens da base CK+ contendo a evolução da expressão facial surpresa. Fonte: Modificada a partir do banco de dados CK+.

base JAFFE foi necessária para se comparar os resultados com o trabalho correlato de (PEDROSO; SALLES, 2012) que utilizou o banco de dados JAFFE.

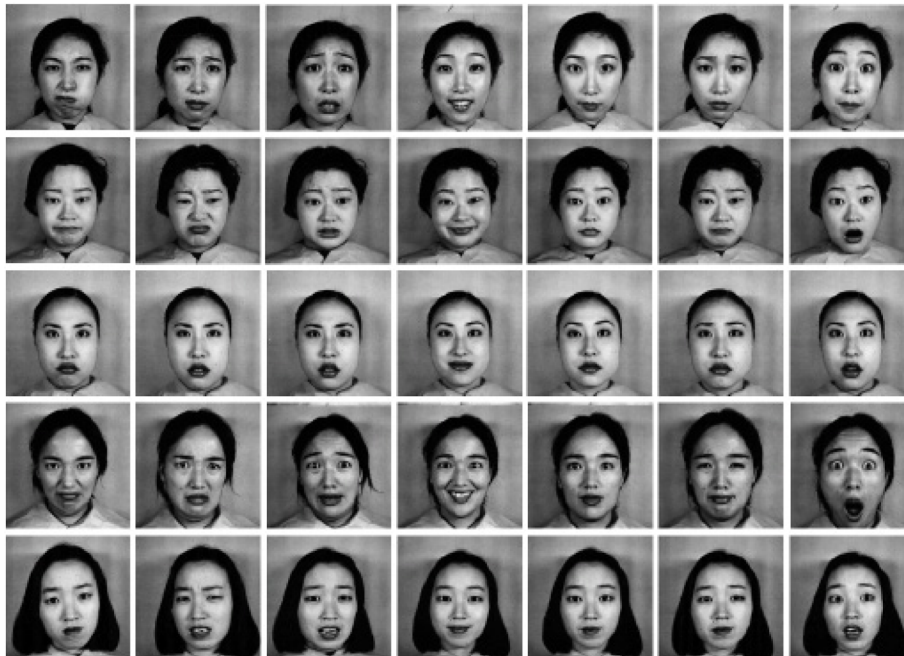


Figura 29 – Banco de dados utilizado: JAFFE - Japanese Female Facial Expression. Fonte: Modificada a partir do banco de dados JAFFE.

4.2 Sistema Proposto

4.2.1 Construção do Modelo AAM

O treinamento do algoritmo AAM para modelagem de uma face para distintos indivíduos em variadas expressões requer um conjunto extensivo de imagens. Esse conjunto deve ser capaz de descrever de maneira mais detalhada possível as múltiplas deformações que uma face pode apresentar, geradas por diferenças de gênero, idade, etnia, pose ou expressões faciais. Tal como discutido na seção 4.1, o banco de dados CK+ engloba em sua base de mais de 10.000 imagens, constituindo diversidade significativa de faces humanas, o que contribui diretamente na qualidade do modelo. Devido a limitações de hardware, das 10.588 imagens da base CK+, 2.000 foram selecionadas aleatoriamente para o treinamento do modelo AAM, dado que para valores superiores a 2.000 o tempo de treinamento se tornava inviável superando 5 horas de processamento, provocando muitas vezes um travamento total do sistema operacional. Outro ponto importante a se destacar é que foram descartados do processo de seleção para treinamento a primeira e a última face de cada subdiretório, uma vez que essas imagens foram escolhidas exclusivamente para o processo de classificação da expressão facial, o que será melhor discutido na próxima seção.

Cada uma das imagens selecionadas para o processo de classificação de expressões faciais foi submetida ao ajuste AAM discutido na Seção 2.2.3 do Capítulo 2, totalizando 1.050 imagens. Portanto, para cada imagem gerou-se um conjunto de 68 *landmarks* ajustadas de modo a marcar pontos específicos da face, mesmo que a face apresente alguma deformação.

Após alguns testes verificou-se que um ponto crítico que influencia na convergência do algoritmo de ajuste AAM é a inicialização da máscara de pontos AAM na face, em relação ao posicionamento e a escala da máscara. Com o propósito auxiliar na convergência do AAM, um procedimento de pré-posicionamento da máscara foi adotado a partir da posição dos olhos. A máscara de pontos é então escalada de maneira que a distância entre a média dos pontos que circundam olhos seja aproximadamente mesma entre os olhos da face teste. A posição dos olhos da imagem de entrada é obtida segundo as marcações *Ground truth* da base CK+. No entanto, essas posições poderiam ser obtidas através de métodos de detecção de olhos, como o de (VIOLA; JONES, 2001).

Posteriormente à etapa de ajuste AAM, um processo de normalização é aplicado às *landmarks* obtidas. Assim sendo, uma transformação escalar foi realizada de forma que a distância entre os olhos seja sempre de 200 pixels.

4.2.2 Extrator de Características

A triangulação de Delaunay, realizada a partir das 68 *landmarks* obtidas com o algoritmo AAM, produz 111 triângulos na face que possuem uma determinada área. Quando unidades de ação são ativadas através do movimento muscular, a posição de determinadas *landmarks* é modificada, uma vez que, o face é deformada. Como resultado, a área de alguns triângulos se altera. Com base na teoria FACS (EKMAN; FRIESEN, 1978), propõe-se que existe uma correlação entre os movimentos da AU's e a variação das áreas dos triângulos de Delaunay. Sendo assim, é possível classificar uma expressão facial baseado nas expressões básicas estudas por Ekman (surpresa, medo, tristeza, felicidade, nojo e nervoso).



Figura 30 – Evolução da expressão facial. Fonte: Modificada a partir do banco de dados CK+.

Considerando a Base CK+, composta por sequências de imagens em seus subdiretórios que se inicia com uma face com expressão neutra evoluindo para uma face com expressão básica, temos uma evolução temporal das áreas, como apresentado na Figura 30. Objetivando a simplificação do método, decidiu-se pela utilização de somente a primeira e a última imagem do subdiretório. Desse modo, observamos que a análise da variação das áreas de forma temporal pode ser realizada subtraindo as áreas dos triângulos correspondentes entre a expressão final (AU's ativadas) e a inicial (neutra), assim para cada um dos 111 triângulos teremos uma variação de área determinada pela ativação de grupos musculares específicos, podendo ser positiva, negativa ou até mesmo neutra.

Desta forma, o conjunto de 111 valores será usado como vetor descritor para o processo de classificação. Portanto, o movimento muscular da face, tanto em intensidade quanto em grupo muscular produzirá um vetor distinto, logo o vetor descritor proposto por esse trabalho guarda grande correspondência com a teoria FACS de Ekman. Ao fazê-lo

dessa forma, acredita-se que é possível tornar a técnica do AAM mais adequada para o problema de reconhecimento de expressões faciais.

O resultado da diferença entre as áreas dos triângulos de uma face com expressão de surpresa e uma face com expressão neutra pode ser observado na Figura 31. Já a Figura 32 demonstra a mesma deformação, embora sendo de outro indivíduo. Por meio da análise comparativa entre os gráficos da Figura 31 e da Figura 32 é possível verificar uma correlação. Já a Figura 33 apresenta a diferença para uma expressão feliz. Apesar de serem do mesmo indivíduo a comparação entre os gráficos da Figura 31 e da Figura 33 apontam para uma menor correlação quando comparado a análise entre a Figura 31 e a Figura 32.

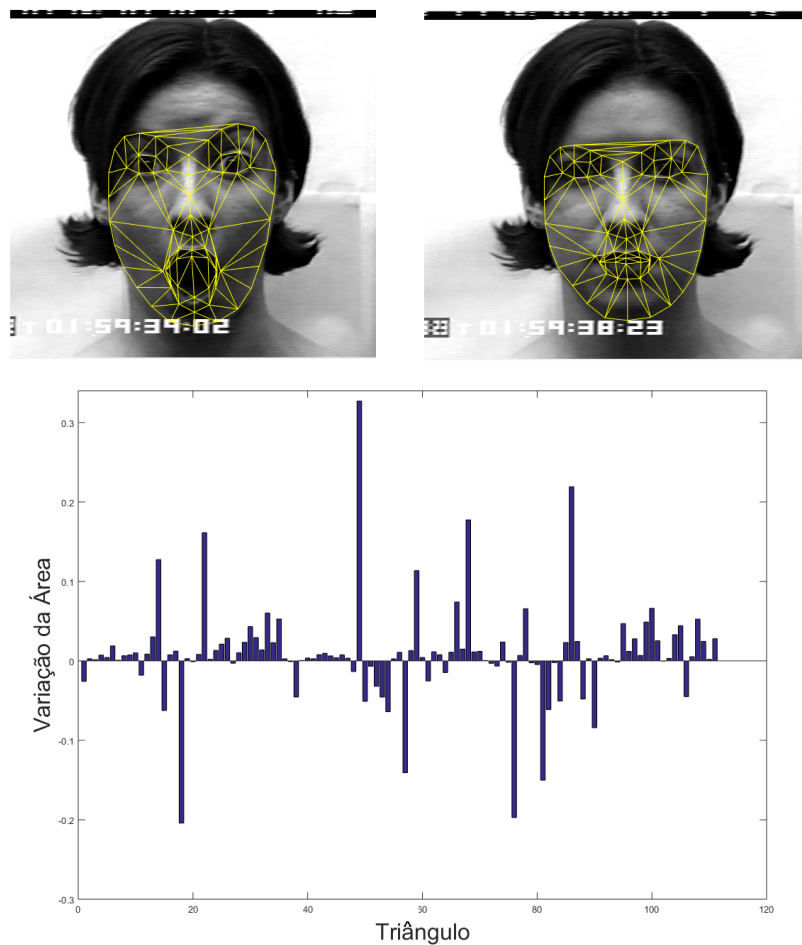


Figura 31 – Gráfico da diferença entre as áreas de uma expressão de surpresa e neutra - Indivíduo 1. Fonte: Modificada a partir do banco de dados CK+.

O processo de extração de características é realizado para cada um dos 525 subdretórios da base CK+ gerando assim o banco de dados de características para o processo de classificação. A validação do modelo de treinamento se dá através do método de *10-fold Cross-Validation* no qual a base de dados é dividida em 10 partes e para cada um dos 10 *folds*, 9 segmentos são selecionados para treinamento e o restante é então usado para validação. A Figura 34 demonstra o método.

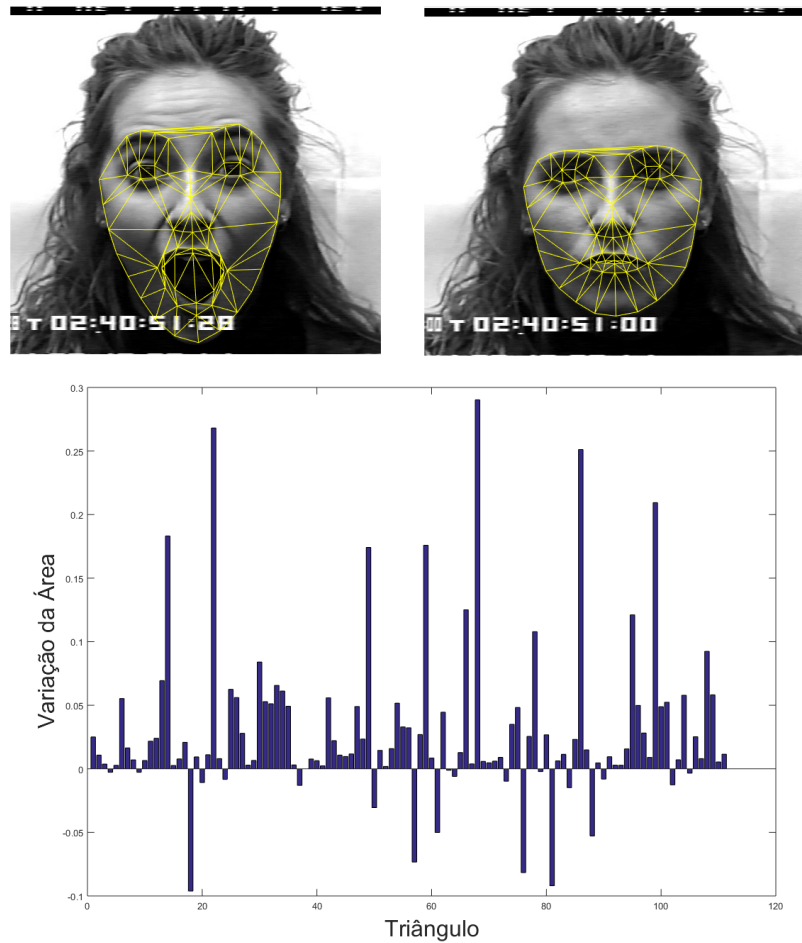


Figura 32 – Gráfico da diferença entre as áreas de uma expressão de surpresa e neutra - Indivíduo 2. Fonte: Modificada a partir do banco de dados CK+.

4.3 Ambiente de Desenvolvimento

A implementação computacional foi realizada em ambiente MATLAB, versão R2015a, pois, entre outros motivos, o MATLAB é uma plataforma que se destaca pela simplicidade com que se pode efetuar operações matriciais e plotagem de gráficos, além de possuir uma grande variedade de ferramentas, classes e funções nativas.

4.3.1 Ajuste dos Classificadores

Para o classificador k-nn a definição do valor mais adequado para k configura outra decisão de grande relevância para o desempenho do método k-nn. Para se determinar o valor de k que apresente melhor desempenho é necessário a realização de experimentos escolhendo diferentes valores para k. Para este trabalho a avaliação de desempenho do classificador k-nn para diversos k se deu através de validação cruzada 10-fold. O número k=5 de vizinhos minimizou o erro do classificador, tendo como medida de similaridade a distância euclidiana. A execução do algoritmo se deu através da função nativa do MATLAB *fitcknn*. A realização do classificador árvore de decisão foi feita através das configurações

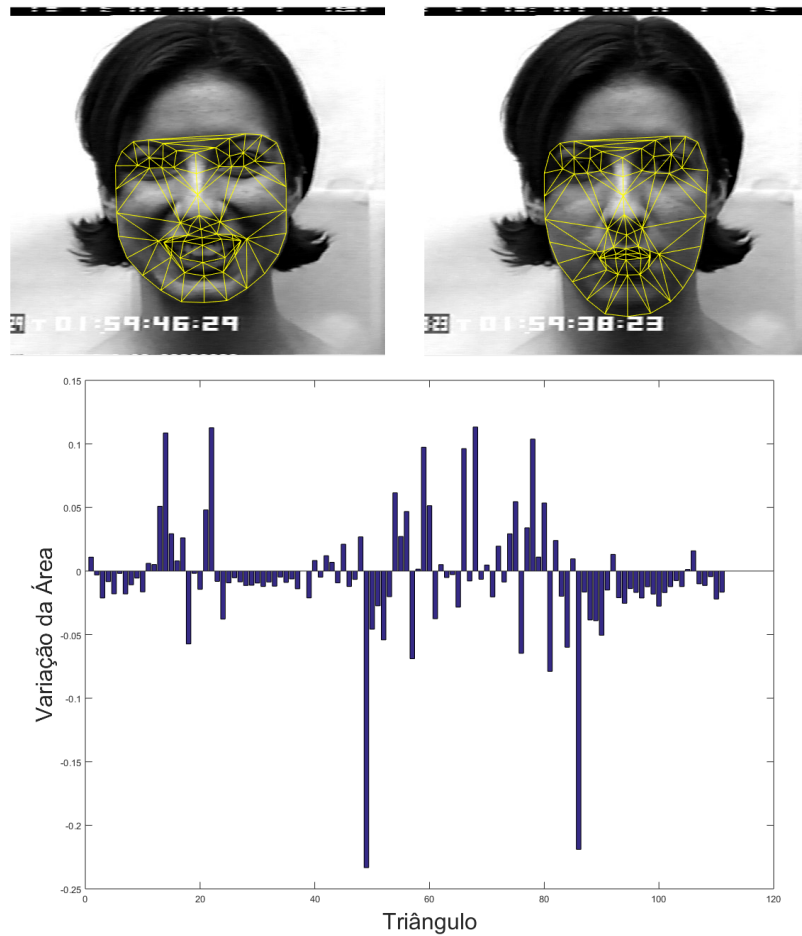


Figura 33 – Gráfico da diferença entre as áreas de uma expressão de feliz e neutra - Indivíduo 1. Fonte: Modificada a partir do banco de dados CK+.

defaults da função *fitctree* que também está presente nas funções nativas do MATLAB.

Já para a Rede Neural o número de camadas e seus respectivos neurônios foram obtidos de forma experimental de modo a se alcançar um bom desempenho na classificação das expressões faciais através da função *train*, contida nas funções nativas do MATLAB. Sendo assim, a rede foi construída com duas camadas ocultas de 50 e 20 neurônios e 6 neurônios na camada de saída, representado na Figura 35.

Devido a característica multi-classe das expressões faciais foi necessário a utilização da biblioteca externa *libsvm*, visto que a função nativa do MATLAB para o classificador SVM trata somente de problemas de duas classes. Sendo assim, a escolha dos parâmetros para a construção do classificador SVM multi-classe se deu de forma experimental buscando-se elevar a acurácia. Portanto, o SVM foi concebido através do kernel linear bem como parâmetro de regularização $C = 20$, parâmetro que controla o custo benefício entre o erro de treinamento e minimização da norma dos pesos.

A Tabela 1 apresenta um resumo das configurações dos classificadores empregados.

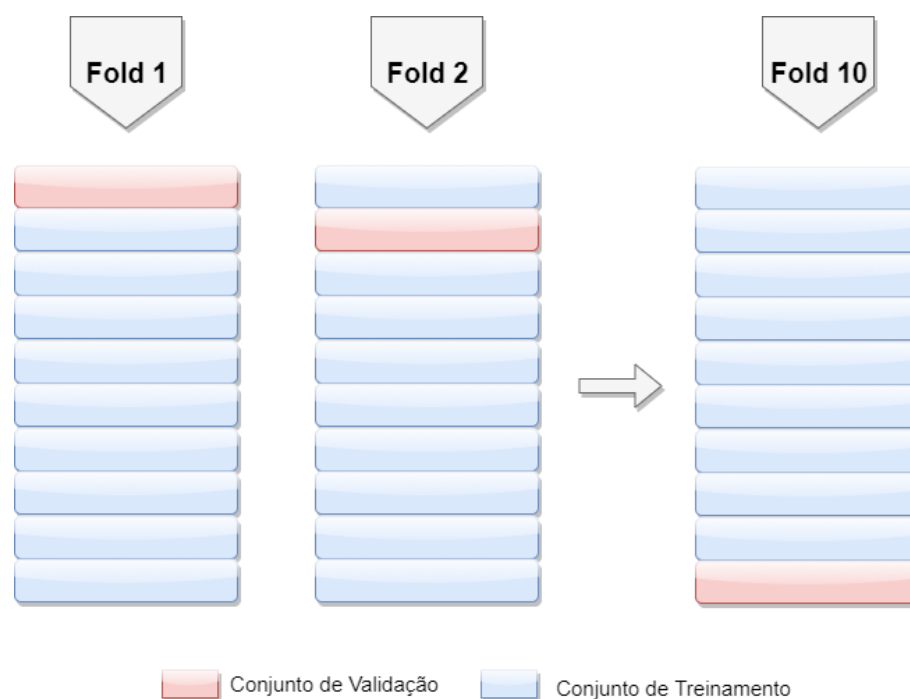


Figura 34 – Representação do método de validação *Cross-Validation* com 10 folds. Fonte: Própria.

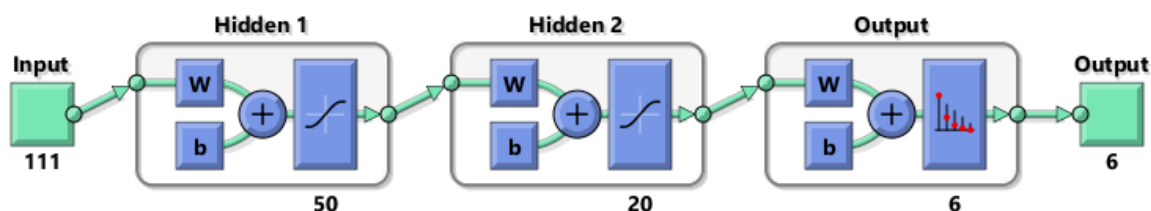


Figura 35 – Representação de uma rede neural com múltiplas camadas. Fonte: Própria.

Classificador	Função	Biblioteca	Configuração
k-nn	<i>fitcknn</i>	MATLAB	k=5
Árvore de decisão	<i>fitctree</i>	MATLAB	<i>default</i> - CART
Rede Neural MLP	<i>train</i>	MATLAB	2 camadas ocultas - (50, 20) neurônios
SVM multi-classe	<i>svmtrain</i>	LIBSVM	Kernel Linear - C=20

Tabela 1 – Configuração dos classificadores.

5 Resultados

5.1 Ajuste AAM

A avaliação da capacidade de generalização do modelo AAM, construído a partir das imagens de treinamento, se deu com base na análise de semelhança das *landmarks* obtidas no ajuste AAM com relação as *landmarks* de referência fornecidas pela base CK+. Para isso, calculou-se o erro médio em pixel das correspondentes marcações, em relação ao *Ground truth* da base CK+. Os resultados do ajuste AAM para as 1.050 faces apresentaram um erro de marcação médio de 3,64 em *pixels*, com valor mínimo de 1,5, Figura 36, e máximo de 15,14 pixels, Figura 37. A Figura 38 exibe o erro médio para cada face.

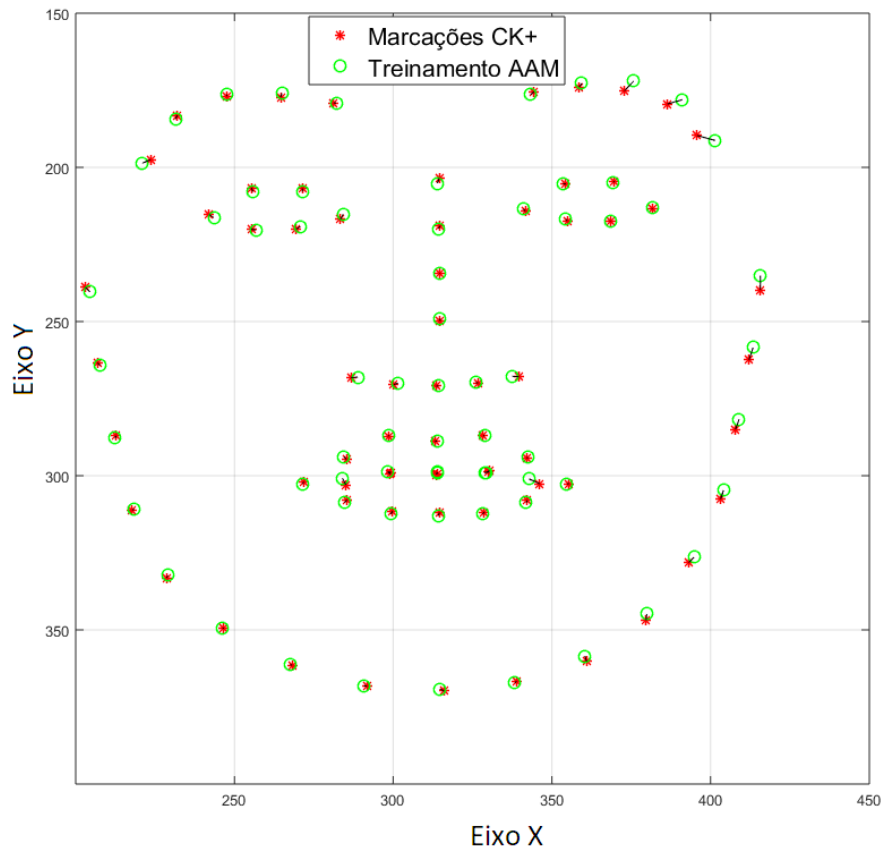


Figura 36 – Melhor resultado AAM - Erro mínimo: 1,5 pixels. Fonte: Própria.

5.2 Extração de Características

Com o objetivo de visualizar a distribuição dos dados e o agrupamento entre as classes, é possível realizar uma projeção dos vetores de descritor obtidos na etapa de extração de características em um espaço cartesiano. A aplicação do PCA, permite

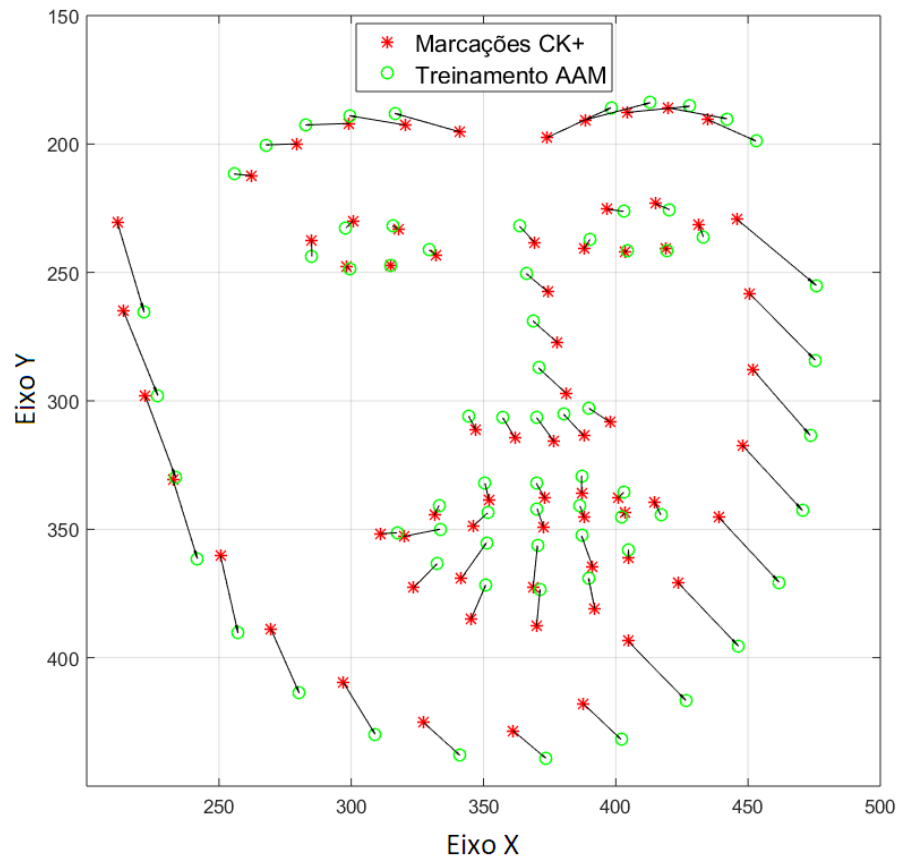


Figura 37 – Pior resultado AAM - Erro máximo: 15,14 pixels. Fonte: Própria.

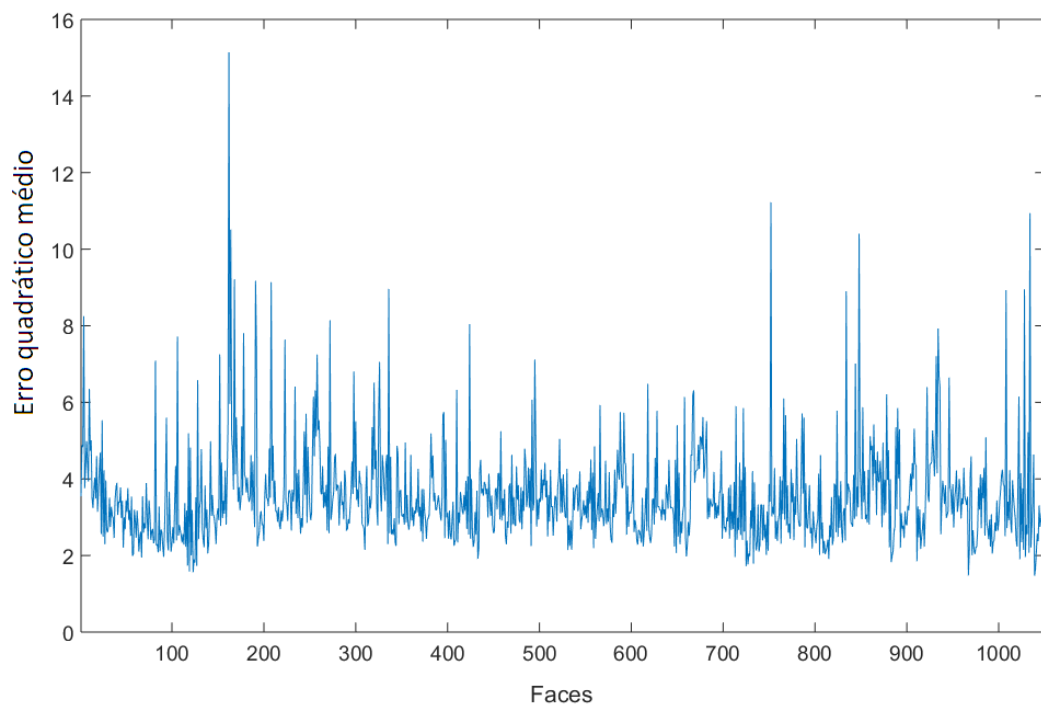


Figura 38 – Erro médio entre as *landmarks*. Fonte: Própria.

reduzir as dimensões (atributos) do vetor descritor, que inicialmente possuía 111 posições, referentes a variação da área dos 111 triângulos, para um vetor com 3 posições, que pode

ser interpretado como a posição x, y, z de um ponto em um espaço vetorial R^3 . A projeção dos 525 subdiretórios analisados na etapa de extração de características é mostrado na Figura 39. . Nesta figura pode-se observar o agrupamento das classes, principalmente as expressões nervoso e triste, que se misturam gerando uma região de maior densidade de classes. Entretanto, é natural que tais expressões se misturem, uma vez que possuem grande correlação, visto que ambas são realizadas por movimentos musculares semelhantes: o decaimento das sobrancelhas. Embora haja regiões onde as classes apresentam menor separação, é possível observar, a partir da Figura 39, que as expressões faciais estão agrupadas de forma que é realizável a separação dos dados a partir do método de extração de caraterísticas proposto.

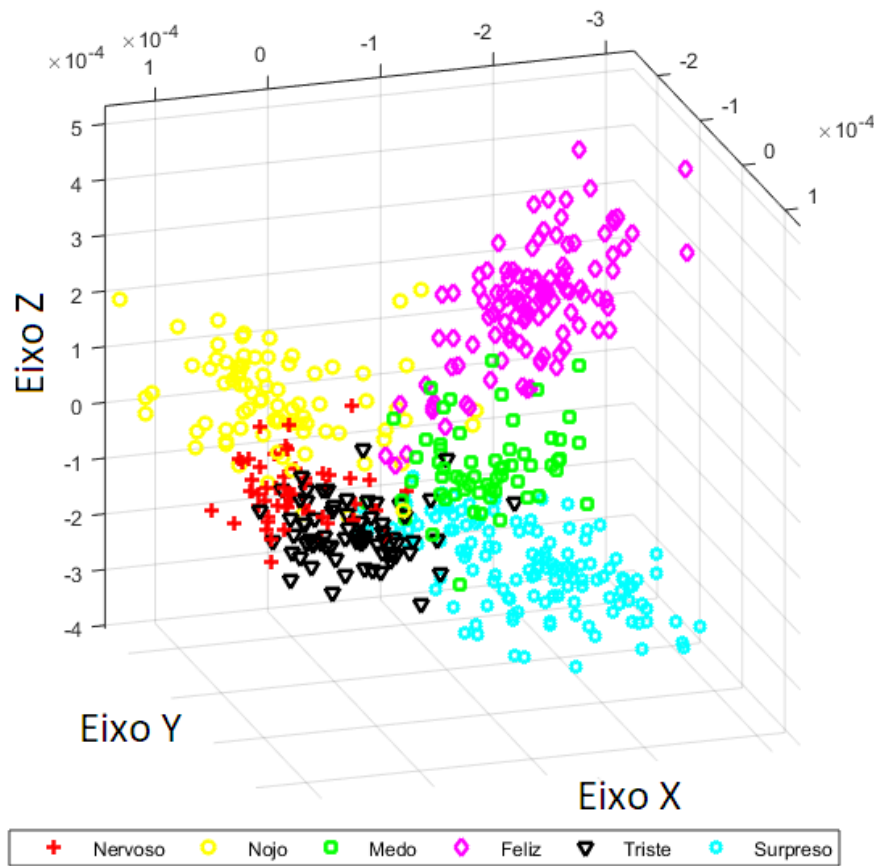


Figura 39 – Distribuição dos dados em 3 dimensões. Fonte: Própria.

5.3 Classificação dos Dados

A Tabela 2 contém a média dos resultados da etapa de classificação das expressões faciais utilizando os classificadores do Capítulo 3 usando validação cruzada com 10 *folds*.

Analizando os resultados obtidos para cada método de classificação, de forma geral, observa-se que o SVM apresentou desempenho superior em relação aos classificadores

Expressão Facial	5-NN	Árvore de Decisão	Rede Neural	SVM
Nervoso	64,51 %	39,47 %	80,7 %	74,8 %
Nojo	86,20 %	60,71 %	82,9 %	89,1 %
Medo	73,93 %	40,90 %	69,5 %	85,6 %
Felicidade	86,66 %	88,00 %	92,7 %	94,8 %
Tristeza	70,00 %	63,63 %	77,1 %	87,7 %
Surpreso	82,05 %	96,49 %	95,1 %	92,7 %
Média:	77,55%	64,86%	83,0 %	87,44%

Tabela 2 – Acurácia por classificador com validação 10-fold - Marcações obtidas a partir do ajuste AAM na base CK+..

	Folds										Média
	1	2	3	4	5	6	7	8	9	10	
Nervoso	87,0	85,0	75,0	83,3	67,0	68,6	70,0	75,0	72,5	65	74,84
Nojo	85,9	87,0	83,3	66,7	86,5	100,0	100,0	100,0	100,0	81,8	89,12
Medo	75,0	72,0	89,0	75,0	100,0	89,3	97,0	82,2	83,4	90,0	85,59
Feliz	85,0	100,0	92,7	93,3	92,3	92,9	98,9	100,0	100,0	92,9	94,8
Triste	66,7	70,1	80,0	92,0	88,0	100,0	100,0	100,0	100,0	79,7	87,65
Surpreso	84,0	96,0	92,3	88,2	97,3	100,0	100,0	90,0	98,0	80,7	92,65
Média	80,6	85,1	85,4	83,0	88,5	91,8	94,3	91,2	92,3	81,7	87,44

Tabela 3 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações obtidas a partir do ajuste AAM na base CK+.

Rede Neural, k-NN e Árvore de Decisão, o que indica que o resultado da classificação das expressões, neste caso, não depende somente das características extraídas da expressão facial, mas também do tipo de classificador utilizado. A Tabela 3 apresentar as taxas de classificação em porcentagem de cada *fold* na validação cruzada para o SVM.

Tal como previsto a partir Figura 39, a taxa de classificação das expressões faciais, principalmente: Surpreso, Felicidade e Nojo é um indicador de que as características utilizadas para descrever as expressões faciais são de fato representativas e se assemelham à forma como o especialista humano classifica visualmente as expressões faciais. Apesar disso, as menores taxas de classificação das expressões Medo, Tristeza e Nervoso, indicam que outras características devem ser integradas ao vetor descritor, de modo a produzir uma maior distinção entre as classes.

Os mesmos processos de extração de características e classificação aplicados às *landmarks* obtidas na etapa de ajuste AAM foram realizados com as *landmarks Ground truth* da base CK+. Os resultados apresentados na Tabela 4, somados aos resultados da Seção 5.1 desse capítulo, referente ao erro quadrático médio entre as marcações Ground

Expressão Facial	5-NN	Árvore de Decisão	Rede Neural	SVM
Nervoso	90,13 %	50 %	77,8 %	94,3 %
Nojo	87,02 %	72,3 %	90,4 %	96,5 %
Medo	68 %	37 %	85,9 %	84,0 %
Felicidade	96,5 %	82,1 %	95,1 %	91,8 %
Tristeza	67,80 %	54,6 %	87,1 %	93,8 %
Surpreso	81 %	87,10 %	88,2 %	94,0 %
Média:	81,7 %	63,85 %	87,4 %	92,4 %

Tabela 4 – Acurácia por classificador com validação 10-fold - Marcações *Ground truth* da base CK+.

	Folds										Média
	1	2	3	4	5	6	7	8	9	10	
Nervoso	100,0	72,0	90,0	100,0	85,8	100,0	95,0	100,0	100,0	100,0	94,3
Nojo	100,0	80,3	90,0	100,0	95,3	100,0	100,0	100,0	100,0	100,0	96,5
Medo	79,0	100,0	88,2	100,0	70,7	85,7	97,4	67,4	66,5	84,9	84,0
Feliz	86,3	100,0	100,0	85,5	88,8	100,0	100,0	82,1	75,2,0	100,0	91,8
Triste	100,0	100,0	85,2	85,7	87,5	88,0	95,7	96,0	100,0	100,0	93,8
Surpreso	100,0	80,0	81,3	100,0	100,0	100,0	95,7	83,2	100,0	100,0	94
Média	94,2	88,7	89,1	95,2	88	95,6	97,3	88,1	90,3	97,5	92,4

Tabela 5 – Acurácia em percentagem (%) a partir da perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações *Ground Truth* da base CK+ com classificador SVM 10-fold.

truth e as marcações obtidas pelo ajuste AAM, reforçam a conclusão de que a modelagem da classe face, através do algoritmo AAM, apresentou rendimento satisfatório no processo de ajuste, uma vez que a diferença apresentada entre os resultados, quando comparado à Tabela 3, é consideravelmente pequena. Contudo, é razoável considerar que a construção do modelo AAM pode ser melhor desenvolvida com objetivo de se atingir o *Ground truth* da base CK+. A Tabela 5 apresenta as taxas de classificação em percentagem de cada *fold* na validação cruzada para o SVM.

5.4 Comparação com outros trabalhos

Como o presente trabalho utilizou o algoritmo AAM no processo de extração de características semelhantemente a (PEDROSO; SALLES, 2012), é relevante comparar os resultados obtidos por ambos. Porém, as etapas de treinamento e validação realizadas por (PEDROSO; SALLES, 2012) foram executadas a partir da base de dados JAFFE citada na Seção 4.1.2 do Capítulo 4. Sendo assim, o mesmo processo de extração de características

aplicado ao banco CK+ foi aplicado para a base JAFFE. Contudo, a base JAFFE não possui *landmarks Ground truth* para a construção do modelo específico AMM e a marcação manual seria um processo dispendioso. Portanto, para demonstrar a generalidade do modelo AAM gerado a partir das faces presentes na base CK+, o mesmo modelo foi utilizado na etapa de ajuste para a base JAFFE, obtendo assim a máscara de pontos ajustada às diferentes expressões contidas no banco de dados. Outro fato relevante, foi a utilização das faces com expressão neutra como referência para o cálculo da diferença das áreas dos triângulos de Delaunay, processo não realizado por (PEDROSO; SALLES, 2012). A projeção dos vetores descritor das expressões analisadas é mostrado na Figura 40, nela é possível observar o mesmo padrão de agrupamento entre as classes da Figura 39, no qual as expressões Nojo, Feliz e Surpreso apresentam maior distinção quando comparadas às expressões Medo, Tristeza e Nervoso.

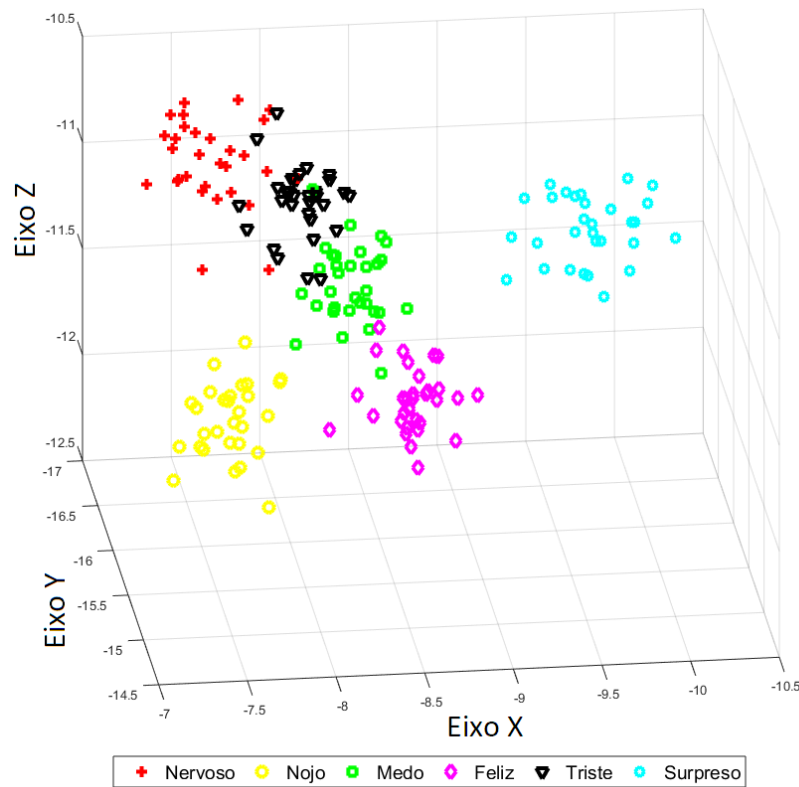


Figura 40 – Projeção dos dados em 3 dimensões - JAFFE Database. Fonte: Própria.

Em (PEDROSO; SALLES, 2012) utiliza-se as *landmarks* obtidas na etapa de ajuste AAM como vetor de características. A escolha desse descritor não possibilitou um bom agrupamento das classes, como pode ser observado na projeção 2D da Figura 40. Outra importante diferença contida em (PEDROSO; SALLES, 2012) é a utilização da expressão neutra como um classe distinta totalizando 7 expressões.

Em relação aos resultados obtidos por (PEDROSO; SALLES, 2012), os autores não apresentaram as taxas de classificação de forma discriminada por expressão, somente disponibilizou a média final. Sendo assim, o melhor resultado obtido foi através do classificador

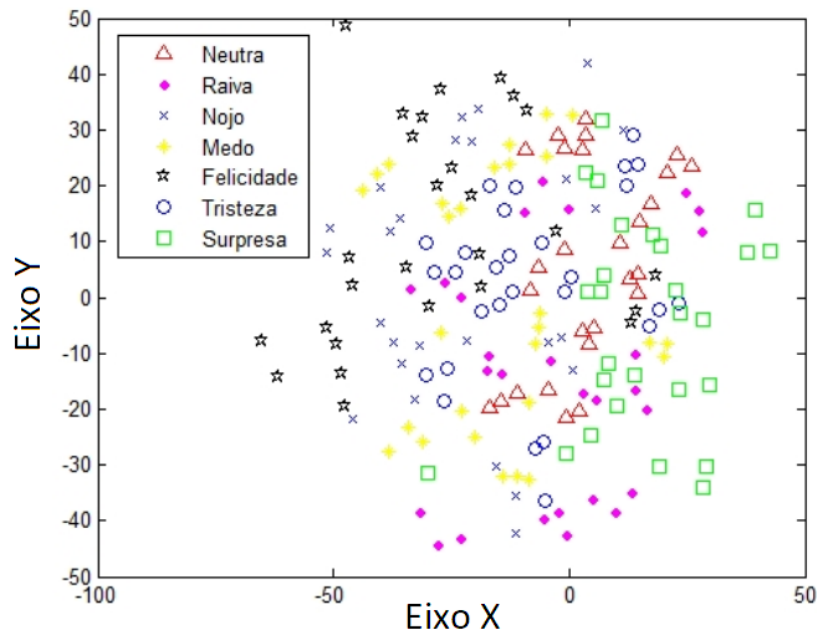


Figura 41 – Projeção dos dados realizada por Pedroso - JAFFE Database - Fonte: (PEDROSO; SALLES, 2012).

SVM-RBF 2-fold com média de acerto de 55,4%. Como era previsto, comparando as projeções de ambos os dados nas Figuras 40 e 41, o método extração de características proposto nesse trabalho alcançou melhores resultados. A Tabela 6 mostra a taxa de classificação para cada expressão, a partir do classificador SVM utilizado nas mesmas configurações aplicadas na classificação do banco de dados CK+. A Tabela 7 apresenta as taxas de classificação em porcentagem de cada *fold* na validação cruzada para o classificador SVM, que apresentou a maior acurácia.

Expressão Facial	5-NN	Árvore de Decisão	Rede Neural	SVM
Nervoso	94,73 %	55 %	92,7 %	85,9 %
Nojo	93,02 %	68,51 %	83,5 %	84,0 %
Medo	75 %	34 %	65,0 %	79,7 %
Felicidade	95,45 %	88,67 %	84,3 %	92,4 %
Tristeza	64,70 %	64,86 %	81,3 %	81,8 %
Surpreso	80 %	90,90 %	89,2 %	94,1 %
Média:	83,82 %	67 %	82,7 %	86,3 %

Tabela 6 – Acurácia por classificador com validação 10-fold - Marcações obtidas a partir do ajuste AAM na base JAFFE.

A Figura 42 apresenta a dispersão de pontos geradas a partir da perspectiva de (PEDROSO; SALLES, 2012) para a construção do descritor da expressão facial utilizando o banco de dados Cohn-Kanade. Aqui, vale destacar sua grande similaridade a dispersão dos dados na Figura 41 apresentada em (PEDROSO; SALLES, 2012). Por meio do classificador

	Folds										
	1	2	3	4	5	6	7	8	9	10	Média
Nervoso	100,0	100,0	63,0	62,5	100,0	82,0	70,5	95,7	85,7	100,0	85,9
Nojo	70,6	80,0	80,0	87,5	100,0	100,0	69,3	83,3	90,0	80,0	84,0
Medo	89,1	75,0	70,0	70,0	70,2	78,0	83,0	74,0	87,5	100,0	79,7
Feliz	95,5	93,3	100,0	100,0	88,1	85,7	89,3	84,6	87,5	100,0	92,4
Triste	100,0	66,7	83,3	67,1	88,1	90,5	76,7	85,0	60,5	100,0	81,8
Surpreso	95,5	80,0	100,0	100,0	100,0	90,0	95,0	85,7	100,0	95,4	94,1
Média	91,8	82,5	82,7	81,2	91,0	87,7	80,6	84,7	85,2	95,9	86.3

Tabela 7 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor da diferença das áreas do triângulos de Delaunay aplicado as marcações obtidas a partir do ajuste AAM à base JAFFE.

SVM, a acurácia obtida foi de apenas 26,5% de acurácia, apresentada de forma detalhada por *fold* na Tabela 8. A acurácia alcançada pelo classificador SVM foi inferior aos 54 % alcançados por (PEDROSO; SALLES, 2012) também obtida por meio do SVM. Entretanto, o padrão dispersivo das classes das Figuras 42 e 41 implica em um considerável esforço do classificador, o que pode gerar situações de *overfitting*, uma vez que os agrupamentos formados apresentam muitas sobreposições, fato semelhante ao ocorrido em (PEDROSO; SALLES, 2012).

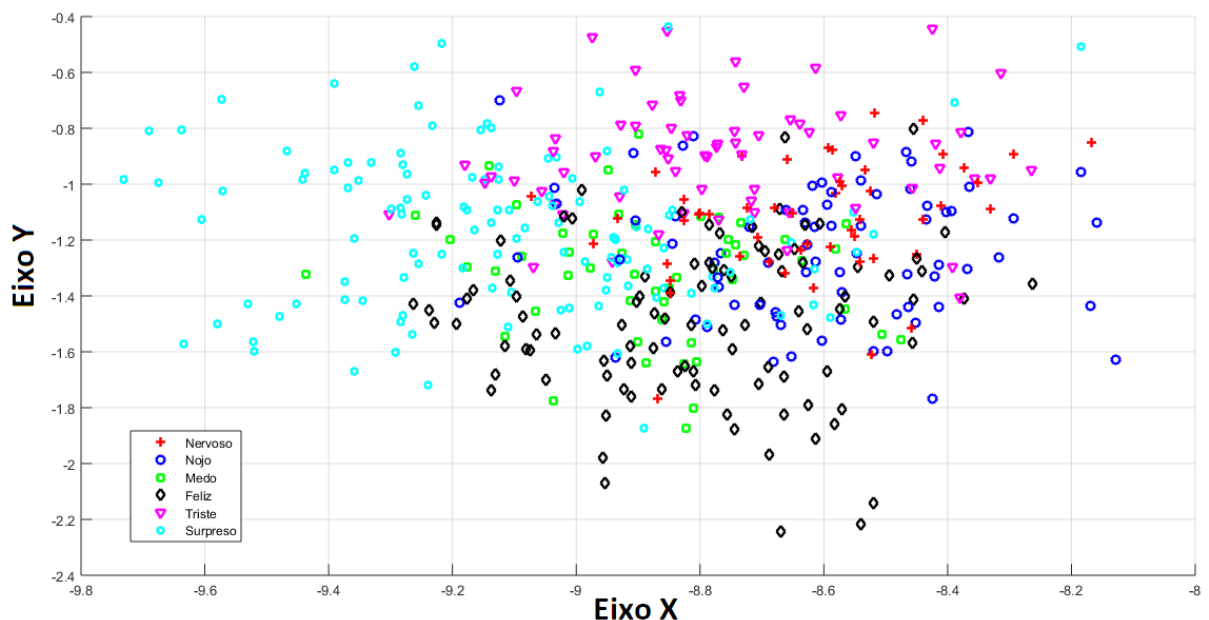


Figura 42 – Projeção dos dados a partir do conceito de descritor de (PEDROSO; SALLES, 2012) para a base CK+. Fonte: Própria.

A Tabela 9 apresenta, para fins de comparação, os resultados alcançados em trabalhos correlatos que utilizaram o banco de dados Cohn-Kanade CK+, porém com outros extratores de características. Comparando os resultados obtidos por (ZHOU; SHI,

	Folds										
	1	2	3	4	5	6	7	8	9	10	Média
Nervoso	25,1	20,1	17,3	30,3	26,1	15,8	10,1	20,0	20,1	22,5	20,7
Nojo	15,7	21,0	19,8	25,8	30,7	24,4	20,2	29,1	22,6	24,4	23,3
Medo	35,1	22,3	13,7	20,1	35,7	30,8	19,7	17,4	22,0	22,3	23,9
Feliz	35,0	31,5	38,8	40,1	22,7	27,4	21,0	30,7	18,0	20,8	28,6
Triste	20,7	21,4	15,7	15,0	20,3	20,7	25,0	29,7	18,0	30,1	21,6
Surpreso	60,0	39,4	35,3	55,7	25,7	31,0	24,4	53,5	49,0	39,4	45,3
Média	31,9	25,9	23,4	31,1	26,8	25,0	20,0	28,5	25,8	26,6	26,5

Tabela 8 – Acurácia em percentagem (%) por fold do classificador SVM com base na perspectiva de descritor de (PEDROSO; SALLES, 2012) aplicado as marcações obtidas a partir do ajuste AAM à base CK+.

2016), (CARCAGNI; DISTANTE, 2015) e (JABID; CHAE, 2010), o método proposto atingiu taxas de classificação menores, superando (PIRES; NETO, 2015). Vale ressaltar que (PIRES; NETO, 2015) somente realizou teste de classificação para a expressão feliz. Outro ponto importante a se destacar, é que a técnica aplicada neste trabalho não utilizou qualquer pré-tratamento na imagem bruta, ou seja, não houve a necessidade de aplicar correção de rotação, escala, corte ou equalização de histograma, ao contrário de todos trabalhos listados acima, que aplicaram todos os métodos mencionados, com exceção de (ZHOU; SHI, 2016) que apenas não realizou a equalização de histograma. Quanto ao método de validação utilizado por cada trabalho citado na Tabela 9, (ZHOU; SHI, 2016), (JABID; CHAE, 2010), (CARCAGNI; DISTANTE, 2015) realizaram validação cruzada *4-fold*, *7-fold*, *10-fold*, respectivamente. Enquanto, (PIRES; NETO, 2015) não menciona nenhum método de validação.

Ano	Autores	Método	Acurácia	Base de Dados
2016	Zhou e Shi	Deep Learning	98,3 %	Cohn-Kanade CK+
2010	Jabid e Chae	LDP e SVM	96,4 %	Cohn-Kanade CK+
2015	Carcagni e Distant	HOG e SVM	95,8 %	Cohn-Kanade CK+
2017	Método proposto	AAM e SVM	92,4 %	Cohn-Kanade CK+
2016	Pires e Neto	LBP e SVM	81,9 %	Cohn-Kanade CK+

Tabela 9 – Resultado de trabalho correlatos.

6 Conclusões

Os resultados apresentados mostram que é possível obter um bom reconhecimento na classificação das expressões faciais em imagens por meio da técnica de modelagem *Active Appearance Model* - AAM, uma vez que o sistema foi capaz reconhecer diferentes expressões para múltiplos indivíduos. Dentre os métodos de aprendizagem de máquina testados, a técnica *Support Vector Machine* (SVM) apresentou os melhores resultados, alcançando valores acurácia de 87,4% para a base CK+ e 86,3% para a banco de imagens JAFFE, seguido pela Rede Neural Multi-camada. Contudo, realizando o mesmo processo de extração de características com as *landmarks* fornecidas pelo banco Cohn-Kanade, a etapa de classificação alcançou acurácia de 92,4% com o classificador SVM o que demonstra que o processo de treinamento e ajuste AAM pode ser melhorado. E embora o classificador k-nn seja considerado um classificador menos robusto em relação ao SVM e Rede Neural, o mesmo atingiu taxas de classificação próximas, corroborando seu desempenho assintótico se comparado a um classificador bayesiano.

A etapa de ajuste AAM apresentou bons resultados, se aproximando das marcações *Ground truth* do banco de dados CK+. O modelo AAM construído mostrou uma capacidade de generalização satisfatória, uma vez que foi possível realizar o processo de ajuste da máscara de pontos em imagens do banco de dados JAFFE. Essa capacidade é essencial para um reconhecimento preciso e correto, uma vez que um sistema de reconhecimento de expressões deve possuir a habilidade de classificar expressões de indivíduos não pertencentes a base de treinamento. Durante o processo de ajuste observou-se que o uso combinado da etapa de ajuste e do pré-posicionamento adequado da máscara AAM na face influencia positivamente na convergência do algoritmo. O aperfeiçoamento da etapa de ajuste proporciona uma base sólida para a etapa de extração de características.

A escolha do método de extração de características, que analisa a evolução expressão facial em relação a face neutra, demonstra um progresso na utilização do AAM em relação a abordagem utilizada por (PEDROSO; SALLES, 2012) na criação do vetor descritor. Em consequência disso, o método proposto nesse trabalho apresentou um avanço nas taxa de classificação, suplantando o método proposto por (PEDROSO; SALLES, 2012) .

O método desenvolvido, além de possuir boa capacidade de identificação de expressões faciais, não exigiu pré-tratamentos nas imagens de entrada, como rotação, equalização de histograma, o que significa vantagem computacional quando comparado a outros métodos. Além disso, a capacidade de generalização da ferramenta AAM possibilita sua aplicação em outros sistemas que necessitem avaliar a deformação de uma classe de objeto.

As *landmarks* ajustadas à face através do ajuste AAM, o cálculo da área dos

triângulos de Delaunay, bem como a escolha de uma análise referenciada à uma face neutra se mostrou uma aproximação eficiente para a classificação de expressões faciais em imagens digitais, em contrapartida ao método prático FACS proposto por Paul Ekman em (EKMAN; FRIESEN, 1978). Senso assim a hipótese levantada no início deste trabalho se mostrou verdadeira e promissora, visto que os movimentos musculares da face, nomeado por Paul Ekman como Unidades de Ação (AUs), puderam ser aproximados e abstraídas através do cálculo da variação espacial dos triângulos de Delaunay referenciados a face neutra. Contudo, ainda existem melhorias a serem desenvolvidas e aplicadas ao método que proposto tanto na etapa de ajuste AAM, quanto a construção do descritor da expressão facial.

6.1 Trabalhos Futuros

A análise da textura de determinadas região através de técnicas como *Histogram of Oriented Gradients* - HOG; Descritores Fractais; *Local Binary Patterns* (LBP), configura uma característica discriminativa para a classificação de expressões faciais. A título de exemplo: uma face com a expressão raiva apresenta um enrugamento da pele na região entre as sobrancelhas; uma expressão de felicidade que devido a exposição dos dentes propicia a alteração de textura na região da boca.

Com objetivo de aprimorar o *framework* AAM, a norma $l2$ pode ser substituída pela norma $l1$ na etapa de ajuste, discutida na Seção 2.2.3 do Capítulo 2, e averiguar se dita substituição melhora a robustez do modelo.

O vetor descritor proposto pode ser incrementado com uma informação mais ampla sobre a evolução dos triângulos de Delaunay, uma vez que a abordagem desenvolvida avalia somente a variação da área dos triângulos. Contudo, um triângulo pode sofrer uma grande deformação, mantendo sua área. É possível presumir que tal alteração no vetor de características influenciará na separabilidade dos dados.

Neste trabalho, não foi discutido a maneira de se obter, de forma automática, a face neutra de uma indivíduo em uma sequência de *frames*, visto que as bases de dados utilizadas já forneciam. No entanto, em sistema real algumas soluções poderiam ser implementadas, como por exemplo: utilizar a forma média do modelo de forma como a face neutra de referência; construir um classificador para identificar somente a face neutra dentre as demais; solicitar ou levar o indivíduo a expressar sua face neutra.

Como apontado anteriormente, o AAM abre um leque para aplicações em diversos problemas de reconhecimento de padrões, devido à sua capacidade de modelar uma classe de objeto e suas possíveis deformações. Sendo assim, o AAM aplicado a um sistema de reconhecimento facial seria capaz de auxiliar na identificação de um indivíduo mesmo que haja deformações devido a pose e expressão facial. Isso se deve ao fato de que, por meio

do AAM, uma face contendo tais características pode ser regredida a sua condição neutra através do método *Piecewise affine warp*, discutido na Seção 2.13 do Capítulo 2. Logo, uma face de um indivíduo em estado neutro poderá ser melhor classificada.

Outra possibilidade é a utilização do AAM em problemas de *tracking*, no entanto, tal aplicação pode exigir um maior poder computacional em virtude do número de *landmarks* necessárias para a construção do modelo, dado que a cada textitframe o processo de ajuste AAM é executado. Dentre outras possibilidades, destaca-se a reconstrução 3D a partir de imagens 2D; leitura labial por intermédio do monitoramento das deformações da boca; a manipulação de máscaras virtuais presentes em aplicativos de celulares e na criação de efeitos especiais no cinema, também conhecido como *live action*.

Referências

- BAKER, S.; MATTHEWS, I. *Lucas-kanade 20 years on: A unifying framework: Part 2*. [S.l.]: The Robotics Institute Carnegie Mellon University: CMU-RI-TR-03-35, 2003. Citado na página 35.
- BIRCK, K. *A Voz do Corpo: A Comunicação Não-Verbal e as Relações Interpessoais*. [S.l.]: XXXI Congresso Brasileiro de Ciências da Comunicação.; 2008 Set 2-6; Natal. RN: Intercom.; 2008. Citado na página 15.
- BREIMAN, L. et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984. Nenhuma citação no texto.
- CARCAGNI, P.; DISTANTE, C. *Facial expression recognition and histograms of oriented gradients: a comprehensive study*. [S.l.]: SpringerPlus, v.4, n.1, p.645, 2015. Citado 2 vezes nas páginas 21 e 63.
- CORTES, C.; VAPNIK, V. Support-vector networks. Kluwer Academic Publishers, Hingham, MA, USA, 1995. Citado na página 42.
- DARWIN, C. *The expression of emotion in man and animals*. [S.l.]: New York: Oxford University Press, 1872. Citado na página 16.
- DAVIES, C. H.; DALTON, C.; CAMPBELL, N. Facial movement based recognition. In: _____. Springer Berlin Heidelberg, 2011. Disponível em: <https://doi.org/10.1007/978-3-642-24136-9_5>. Citado na página 16.
- DONATO M. BARTLETT, J. H. P. E. T. S. G. *Classifying facial actions*. [S.l.]: IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 21, n. 10, p. 974 –989, 1999. Citado na página 21.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Ed)*. [S.l.]: Wiley, 2000. Citado na página 40.
- DYER, G. G. C. *Learning from examples in the small sample case: face expression recognition*. [S.l.]: IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, v. 35, n. 3, p. 477 –488, 2005. Citado na página 21.
- EKMAN, P.; FRIESEN. *Constants across cultures in the face and emotion*. [S.l.]: Personality Social Psychol, 17(2):124-129, 1971. Citado na página 16.
- EKMAN, P.; FRIESEN. *Facial action coding system*. [S.l.]: Personality Social Psychol, 1978. Citado 4 vezes nas páginas 16, 17, 50 e 65.
- FARIA RODRIGO A. M. BRAGA, E. V. P. M. *Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions*. [S.l.]: Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium, 2007. Citado na página 16.
- FUJIMORA, M. S. N. S. K. A. *A preliminary note on pattern recognition of human emotional expression*. [S.l.]: International joint conference on pattern recognition., 1978. Citado na página 21.

- FURNHAM, A. *Linguagem corporal no trabalho – Você S.A.* [S.l.]: Nobel, 2001. (Exame Você S.A). Citado na página 14.
- GROSS, I. M. R.; BAKER, S. *Generic vs. person specific active appearance models*. [S.l.]: IVC vol 23 pp 1080-1093, 2005. Citado na página 36.
- HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. IEEE Press, 1994. Citado na página 42.
- HAYKIN, S. S. *Neural networks and learning machines*. Third. [S.l.]: Pearson Education, 2009. Citado 3 vezes nas páginas 8, 40 e 41.
- IZARD, C. *Emotion Knowledge as a Predictor of Social Behavior and Academic Competence in Children at Risk*. [S.l.]: University of Delaware, 2001. Citado na página 14.
- JABID, T.; CHAE, O. *Robust Facial Expression Recognition Based on Local Directional Pattern*. [S.l.]: ETRI Journal, Volume 32, Number 5: 784 - 794, 2010. Citado 2 vezes nas páginas 22 e 63.
- KANADE, J. C. T.; TIAN, Y. *Comprehensive Database for Facial Expression Analysis*. [S.l.]: IEEE Int. Conf. Autom. Face Gesture Recognition: p. 46-53, 2000. Citado na página 47.
- KUNG, M. A. Z. S. H.; BOUCHAFFRA, D. 3d hmm-based facial expression recognition using histogram of oriented optical flow. Transactions on Machine Learning and Artificial Intelligence Volume 3, Issue 6, 2015. Citado na página 22.
- LUCEY, P. et al. *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression*. [S.l.]: <http://www.pitt.edu/emotion/ck-spread.html> - Acessado em 10/06/2016, 2010. Citado 2 vezes nas páginas 24 e 47.
- LYONS, M.; KAMACHI, M.; GYOBA., J. *Japanese Female Facial Expression (JAFFE) Database*. [s.n.], 2001. Disponível em: <https://figshare.com/articles/jaffe_desc_pdf/5245003>. Citado na página 47.
- MANGASARIAN, O. L.; MUSICANT, D. R. Lagrangian support vector machines. *J. Mach. Learn. Res.*, JMLR.org, 2001. Citado na página 42.
- MARTIN, C.; WERNER, U.; , H. M. G. . *A real-time facial expression recognition system based on active appearance models using gray images and edge images*. [S.l.]: Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition, 2008. Citado na página 22.
- MATTHEWS, I.; BAKER, S. *Active Appearance Models Revisited*. [S.l.]: IJCV - International Journal of Computer Vision: 60(2) 135-164, 2004. Citado 6 vezes nas páginas 8, 24, 31, 34, 35 e 36.
- MOON, H.; SHARMA, R.; JUNG, N. *Method and system for measuring shopper response to products based on behavior and facial expression*. [S.l.]: ICAP Patent Brokerage, 2012. Citado na página 16.
- OMLIN, J. W. C. *Haar features for faces au recognition*. [S.l.]: 7th International Conference on Automatic Face and Gesture Recognition. p. 5 pp. – 101, 2006. Citado na página 21.

- PANTIC, M. et al. Web-based database for facial expression analysis. p. 317–321, 2005. Citado na página 22.
- PEDROSO, F.; SALLES, E. *Reconhecimento de expressões faciais baseado em modelagem estatística*. [S.l.]: CBA - Congresso Brasileiro de Automatica: 631-638, 2012. Citado 11 vezes nas páginas 9, 10, 22, 46, 48, 59, 60, 61, 62, 63 e 64.
- PIRES, A.; NETO, G. *Compound Local Binary Pattern para reconhecimento de expressões faciais*. [S.l.]: Universidade Federal do Maranhão, São Luís, Brasil, 2015. Citado 2 vezes nas páginas 22 e 63.
- REEVES, D. B. *The Learning Leader: How to Focus School Improvement for Better Results*. [S.l.]: ASCD; 1º Ed., 2006. Citado na página 14.
- RUSSELL, J.; FERNÁNDEZ-DOLS, J. *The Psychology of Facial Expression*. [S.l.]: Cambridge University Press, 1997. Citado na página 16.
- SEWELL, M. Principal component analysis. CiteSeer, 2007. Citado 2 vezes nas páginas 71 e 75.
- SILVA, G. Estudo da técnica pca (análise de componentes principais) e auto-faces aplicadas ao reconhecimento de faces humanas. Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos., 2005. Citado na página 71.
- THEODORIDIS, S.; KOUTROUMBAS, K. Pattern recognition. Academic Press 4th., 2009. Citado 5 vezes nas páginas 8, 9, 43, 45 e 46.
- TZIMIROPOULOS, G.; PANTIC, M. *Optimization problems for fast AAM fitting in-the-wild*. [S.l.]: ICCV IEEE 593-600, 2013. Citado 2 vezes nas páginas 36 e 37.
- VAPNIK, V. *Inductive Principles of the Search for Empirical Dependences (Methods Based on Weak Convergence of Probability Measures)*. [S.l.]: Morgan Kaufmann, 1989. Citado 2 vezes nas páginas 42 e 44.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: . [S.l.: s.n.], 2001. p. 511–518. Citado na página 49.
- ZHOU, Y.; SHI, B. E. *Action Unit Selective Feature Maps in Deep Networks for Facial Expression Recognition*. [S.l.]: Workshop on Neuroinformatics and Neurobiology - NBNI: 185-191, 2016. Citado 2 vezes nas páginas 21 e 63.

Apêndices

APÊNDICE A – PCA - Análise de Componentes Principais

O PCA - *Principal Components Analysis*, ou em tradução direta, Análise dos componentes principais, sendo desenvolvido inicialmente por Pearson em 1901 e aprimorado posteriormente por Hotelling em 1933. O PCA é um método que tem por finalidade básica representar dados em uma nova base visando a redução de dimensionalidade. Armazenando os dados mais representativos a partir de combinações lineares das variáveis originais (SILVA, 2005). A PCA é um dos métodos estatísticos de múltiplas variáveis mais simples, considerado uma transformação linear ótima, sendo amplamente utilizada pela comunidade de reconhecimento de padrões. Em problemas de classificação o PCA é muito útil quando os vetores de descritores que descrevem uma instância de classe possui muitas dimensões tornando seu processamento dispendioso, proporcionando uma redução do custo computacional. No qual sempre é desejável dispor de uma representação compacta e de um bom poder de discriminação de classes de padrões. Para isso, é importante que não haja redundância entre as diferentes características dos padrões, ou seja, que não haja covariância entre os vetores da base do espaço de características

Cada vetor de características com dimensionalidade reduzida é obtido a partir da combinação dos n maiores autovetores ponderados por pesos específicos. Sendo n o número dos mais representativos autovetores obtidos de acordo com os autovalores correspondentes ordenados de forma decrescente. A escolha de n define o grau de compressão do método. Contudo, sua escolha deve ser feita considerando a representatividade aceitável dos dados.

A ideia de compressão ou de redução de dimensionalidade se baseia o na perspectiva de que projeções lineares sejam capazes de conservar o essencial da informação envolvida (SEWELL, 2007). Sempre real e simétrica, a matriz de covariância obtida a partir dos dados, fornecerá o conjunto de autovalores e seus correspondentes autovetores ortonormais. Cada autovetor selecionado, disposto em coluna, forma a matriz de transformação utilizada para o cálculo do PCA.

Outra importante aplicação do PCA é representação gráfica da dispersão dos dados em espaço R^1 , R^2 ou R^3 , mesmo que os dados originais possuam dimensão superior. Essa representação é pode ser muito útil para se entender a relação entre as variáveis e seus agrupamentos. A Figura 43 exibe uma representação gráfica do resultado da compressão de dados após a aplicação do PCA. Sendo a Figura 43 a) a representação dos dados em um espaço de R^3 e a Figura 43 b) a projeção dos dados na nova base R^2 criada a partir dos 2 autovetores com maior energia. A Figura 44 resume do processo necessário para a

aplicação da método PCA a um conjunto de dados.

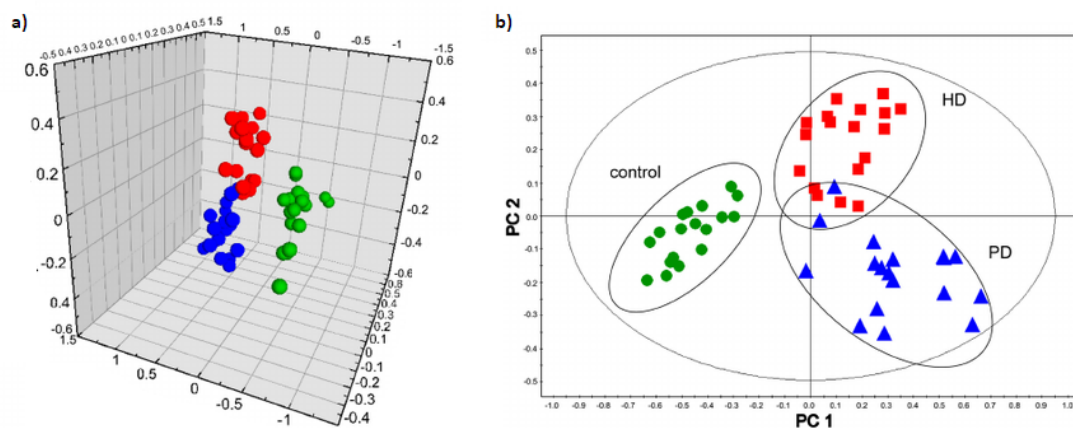


Figura 43 – Demonstração gráfica da redução de dimensionalidade através da PCA

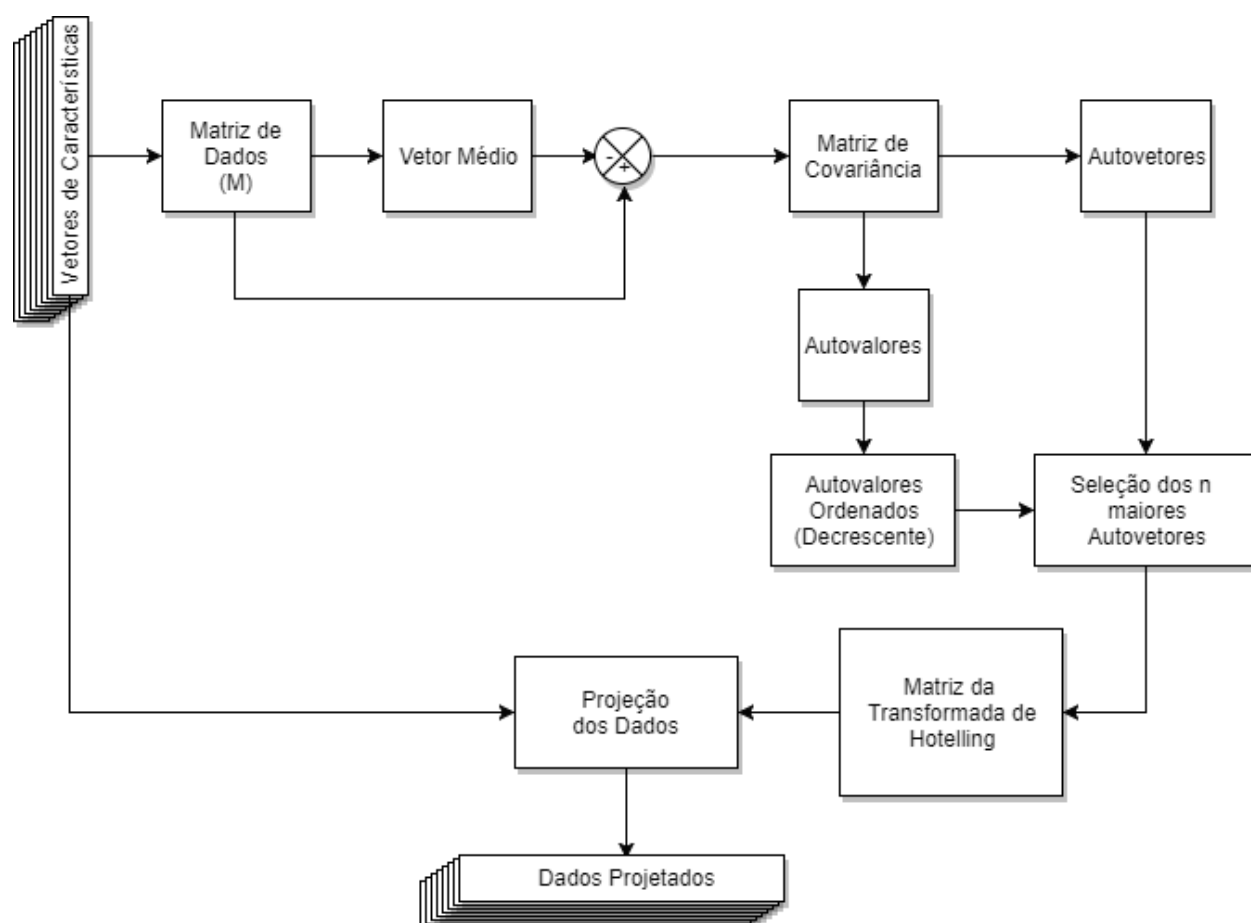


Figura 44 – Esquema das etapas do PCA

A.1 Matriz de Covariância

Em estatística, muitas aplicações calculam a matriz de covariância para os estimadores de parâmetros em um modelo estatístico. Todas as matrizes de covariância são positivas semi definidas simétrica e sumarizam a covariância entre n variáveis. Matematicamente, a matriz de covariância é definida por:

$$\Sigma = E \left[(\mathbf{X} - \mathbf{E}[\mathbf{X}])(\mathbf{X} - \mathbf{E}[\mathbf{X}])^T \right]. \quad (\text{A.1})$$

Sendo \mathbf{X} um vetor coluna de variáveis aleatórias, como:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_3 \end{bmatrix}, \quad (\text{A.2})$$

e E o valor esperado. A matriz de covariância pode ser calculada de forma sistemática com base nas seguintes etapas:

O cálculo do vetor médio de \mathbf{X} é o obtido através de

$$m_{\mathbf{X}} = \frac{1}{M} \sum_{i=1}^M x_i, \quad (\text{A.3})$$

no qual x_i representa cada amostra da variável aleatória \mathbf{X} . A título de exemplo, consideremos as seguintes amostras x_i :

$$x_1 = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad (\text{A.4})$$

$$x_2 = \begin{bmatrix} 5 \\ 6 \\ 6 \\ 1 \end{bmatrix}, \quad (\text{A.5})$$

$$x_3 = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 7 \end{bmatrix}, \quad (\text{A.6})$$

O vetor médio destas amostras será:

$$m_{\mathbf{X}} = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 4 \end{bmatrix}, \quad (\text{A.7})$$

A próxima etapa na construção da matriz de covariância é a subtração entre \mathbf{X} e $m_{\mathbf{X}}$ de modo:

$$x_1 - m_{\mathbf{X}} = \begin{bmatrix} -1 \\ -2 \\ -2 \\ 0 \end{bmatrix}, \quad (\text{A.8}) \quad x_2 - m_{\mathbf{X}} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ -3 \end{bmatrix}, \quad (\text{A.9}) \quad x_3 - m_{\mathbf{X}} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 4 \end{bmatrix}, \quad (\text{A.10})$$

O produto de cada resultado por sua transposta resultará cada um em uma matriz 4x4, dada por $(x_i - m_{\mathbf{X}})(x_i - m_{\mathbf{X}})^T$. No exemplo em questão teremos as seguintes matrizes respectivamente:

$$(x_1 - m_{\mathbf{X}})(x_1 - m_{\mathbf{X}})^T = \begin{bmatrix} 9 & 6 & 6 & 0 \\ 6 & 4 & 4 & 0 \\ 6 & 4 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{A.11})$$

$$(x_2 - m_{\mathbf{X}})(x_2 - m_{\mathbf{X}})^T = \begin{bmatrix} 4 & 4 & 2 & -6 \\ 4 & 4 & 2 & -6 \\ 2 & 2 & 1 & -3 \\ -6 & -6 & -3 & 9 \end{bmatrix}, \quad (\text{A.12})$$

$$(x_3 - m_{\mathbf{X}})(x_3 - m_{\mathbf{X}})^T = \begin{bmatrix} 1 & 0 & -1 & -4 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 4 \\ -4 & 0 & 4 & 16 \end{bmatrix}, \quad (\text{A.13})$$

Sendo assim, a matriz de covariância para M amostras pode ser calculada a partir de:

$$C_{\mathbf{X}} = \frac{1}{M} \sum_{i=0}^M (x_i - m_{\mathbf{X}})(x_i - m_{\mathbf{X}})^T. \quad (\text{A.14})$$

Tendo como base os resultados anteriores, obtemos a seguinte matriz:

$$C_{\mathbf{X}} = \begin{bmatrix} 14/3 & 10/3 & 7/3 & -10/3 \\ 10/3 & 8/3 & 2 & -2 \\ 7/3 & 2 & 2 & 1/3 \\ -10/3 & -2 & 1/3 & 25/3 \end{bmatrix}, \quad (\text{A.15})$$

O exemplo de pode ser facilmente estendido para o cálculo da matriz de covariância de uma vetor coluna de u com de qualquer número de amostras. Se tais amostras apresentarem 4 dimensões a matriz $C_{\mathbf{X}}$ será 4x4, se possuírem 5 dimensões a matriz será 5x5, e caso tenham dimensão N a matriz de covariância terá dimensão $N \times N$.

A.2 Autovalores e Autovetores

Um escalar λ é um autovalor da matriz quadrada A , se existir um vetor v não nulo, que satisfaça a igualdade

$$A.v = \lambda.v, \quad (\text{A.16})$$

tal vetor é chamado de autovetor de A associado a um λ . Logo, a igualdade $A.v = \lambda.v$ pode ser reescrita como

$$(A - \lambda.I)v = 0, \quad (\text{A.17})$$

sendo I uma matriz identidade. Desta forma, λ e v pode ser encontradas a partir de

$$\det(A - \lambda.I) = 0. \quad (\text{A.18})$$

A.3 Autovalores e Autovetores

Os autovetores e autovalores associados a matriz de covariância calculada na seção anterior são mostrados abaixo, respectivamente:

$$v = \begin{bmatrix} 0,1089 & 0,7254 & -0,3864 & -0,5591 \\ 0,5992 & -0,5904 & -0,3732 & -0,3914 \\ -0,7626 & -0,3131 & -0,5348 & -0,1854 \\ 0,2179 & 0,1648 & -0,6523 & 0,7070 \end{bmatrix}, \quad (\text{A.19})$$

$$\lambda = \begin{bmatrix} 0 & 0,1897 & 5,4880 & 11,9889 \end{bmatrix}. \quad (\text{A.20})$$

A.4 Matriz de Transformação de Hotteling

Cada autovalor representa a variância explicada para cada componente principal. Logo, a soma dos autovalores da matriz de covariância é chamado de traço e representa a variância total dos dados. Sendo a distância média quadrada entre cada amostra e o centro no espaço n -dimensional. A soma dos primeiros n autovalores é proporcional ao total de variância explicada pelas primeiras k -dimensões do PCA (SEWELL, 2007). Novamente

recorrendo ao exemplo anterior, dado os autovalores λ associados a C_X , temos que o traço igual a:

$$trace = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 0 + 0,1897 + 5,4880 + 11,9889 = 17,6666. \quad (A.21)$$

De maneira que os eixos correspondentes aos autovalores λ_1 e λ_2 somados, representam apenas 1,07% de informação discriminante dos dados. Enquanto os eixos correspondentes aos autovalores λ_3 e λ_4 compreendem 98,93% da informação. Portanto, a seleção dos autovetores associados aos autovalores λ_3 e λ_4 produz a seguinte matriz de Hotelling W :

$$W = \begin{bmatrix} -0,5591 & -0,3864 \\ -0,3914 & -0,3732 \\ -0,1854 & -0,5348 \\ 0,7070 & -0,6523 \end{bmatrix}, \quad (A.22)$$

A projeção das amostras x_1 , x_2 e x_3 na base criada através do PCA é apresentada a seguir:

$$x'_1 = \begin{bmatrix} 2,8308 \\ 2,9750 \end{bmatrix}, \quad (A.23) \quad x'_2 = \begin{bmatrix} -4,2074 \\ -0,0969 \end{bmatrix}, \quad (A.24) \quad x'_3 = \begin{bmatrix} 3,2018 \\ -2,7576 \end{bmatrix}, \quad (A.25)$$

Analisando os dados alcançados, é concludente a capacidade de compressão do PCA, uma vez que em relação as amostras iniciais, houve uma redução de 50% do vetor x_i , ainda assim, mantida 98,3% da representatividade dos dados. A Análise de Componentes Principais será uma ferramenta imprescindível para o desenvolvimento das etapas posteriores deste trabalho.